DHT Performance for Peer-to-Peer SIP – A Mobile Phone Perspective

Imre Kelényi
Budapest University of Technology and Economics
Budapest, Hungary
imre.kelenyi@aut.bme.hu

Jukka K. Nurminen
Nokia Research Center
Helsinki, Finland
jukka.k.nurminen@nokia.com

Marcin Matuszewski
Future Invest
Warsaw, Poland
marcin.matuszewski@futureinvest.pl

***Abstract*—In this paper we investigate the DHT performance for Peer-to-peer SIP. We analyze performance from the mobile phone perspective. This angle brings battery consumption and efficient use of the limited resources of the mobile devices to the focus. We develop an analytical model of Kademlia DHT performance and use it to analyze a number of different strategies to use peer-to-peer SIP with mobile devices. These analyses include studying a model where all devices participate in the DHT and contrasting that with a model where only a subset of devices forms the DHT. Our analysis shows that in a one million user P2PSIP network, at least 30% of the mobile nodes should participate in operating the DHT to remain energy-efficient.**

Keywords-mobile, peer-to-peer, DHT, SIP, energy-efficiency

I. INTRODUCTION

Peer-to-peer SIP (P2PSIP), which is currently in the standardization process, is looking for ways to implement the Session Initialization protocol (SIP) in a peer-to-peer fashion. SIP is a signaling protocol used for setting up multimedia communication sessions, such as voice and video calls, over the IP networks, in particular the Internet. SIP uses a number of centralized servers, such as proxies and registrars, that have to be maintained by a service provider.

P2PSIP replaces the fixed hierarchy of the client-server SIP with a P2P overlay network. The main task of this overlay is to provide a distributed storage mechanism that allows mapping names to network locations, enabling contact information storage, service publication and discovery. Another task of the overlay is to provide services, such as NAT (Network Address Translation) traversal or presence services, in a distributed fashion.

The overlay consists of nodes called P2PSIP peers, which communicate using P2PSIP peer protocol, such as P2PP, and run collectively a distributed database algorithm, and store data on behalf of other nodes or users. The algorithm could be realized using a distributed hash table (DHT). A second node-type, P2PSIP client, connects to one or more peers using a P2PSIP client protocol and allows P2PSIP applications to access the distributed database. In contrast with peers, clients do not run a distributed database algorithm. The decision whether a device should become a peer or a client could be done based on the available resources, such as transmission speed or storage capacity, service operator policy, or battery consumption.

Since mobile phones are widely used for real-time communication, such as voice calls, it is important to understand how the P2PSIP would work on the mobile phone hardware. One particularly important aspect of mobile use is the energy consumption of the applications. If an application rapidly drains the battery of the mobile device, it is not likely to get user acceptance. Therefore special analysis of the energy consumption of any proposed mobile phone application is important.

Real-time services such as VoIP require high reliability and non-stop connectivity to the IP network. This means that if mobile phones provide P2PSIP peer functionality, the P2PSIP software needs to be always connected to the P2PSIP overlay. The power consumption in P2PSIP may be impacted by maintenance of the routing state and by the handling of lookups and requests to store data on behalf of other users or nodes. These two factors mainly affect the idle state, when the P2PSIP software runs on a battery-powered device but no services provided by the overlay are used at the moment. The power consumption in the idle state determines how long a battery-powered device can stay online without recharging. In addition to being connected to the overlay, all of the overlay functions require transmission of data over the expensive air interface. For this reason the P2PSIP overlay protocols must also be optimized to reduce potential charges incurred by transmitting data over the air interface and to limit the overall traffic in the mobile networks that may have impact on the overall QoS.

Furthermore, we have to remember that typically mobile phones are behind NATs, which makes them hard to be accessible to other network nodes. There are mechanisms to traverse NATs such as one presented in [1], but these mechanisms increase traffic on the air interface and consumption of battery power. Most P2P algorithms require connectivity to more than one node. Typically the more connections a mobile device needs to handle the higher the energy consumption is and the more traffic is sent over the air interface.

In [2] the energy consumption behavior of a Kademlia DHT implementation on mobile phones is investigated. This paper complements that research by taking the typical traffic patterns of SIP into account and viewing the results through the lens of P2PSIP needs. In particular, we focus on the role of the mobile devices in the DHT. Kademlia was chosen as the DHT protocol of our analysis since currently this is the only protocol that is used in deployed networks with more than one million users: the Mainline BitTorrent DHT and the Azureus DHT [3].

The main contributions of this paper are:

- Analytical DHT performance model which emphasizes the aspects that are essential for P2PSIP application and mobile device use
- Application of the model to the P2P SIP use cases and quantitative results that predict the performance of P2PSIP in different network sizes
- Comparison of different strategies to implement P2PSIP with DHT

The rest of the paper is structured as follows. In Section II, we derive a performance model of the Kademlia DHT. In Section III, we apply the model to peer-to-peer SIP and compare different implementation strategies. In Section IV, we compare our work with related research, and finally, in Section V, we summarize our conclusions.

II. MODEL ON P2P-SIP MESSAGE OVERHEAD

In this section we define a model on the performance of the Kademlia DHT and the amount of traffic a P2PSIP node experiences. The first step is determining the message overhead which is the number of messages sent during performing a DHT operation. As a basis we use an existing general DHT model presented by [4]. However, it is neither SIP nor Kademlia specific, so we extend it to support the different SIP related messages and also take into account the routing and bucket mechanism of Kademlia. Furthermore, we use our earlier measurement results of Mainline BitTorrent DHT [2] to map the variables of the model to the energy consumption of a mobile device. Table I. shows the notation used in the model.

TABLE I. NOTATION

Symbol	Definition
n	the number of service nodes (peers) in the DHT
c	the number of clients
p	the DHT churn rate: the probability that a message is sent to an active contact
φ	number of calls initiated by a single client in one second
T_{online}	the average client online time (how many seconds a client is online), used for calculating the client registration traffic
T_t	the stored value update frequency in seconds(global Kademlia parameter)
T_{rfs}	bucket refresh timeout (if a bucket in a node's routing table has not been updated for T_{rfs} seconds, then a bucket refresh operation is performed)
k	the Kademlia bucket size (in Mainline BitTorrent k=8)
$E[C]$	message overhead: the number of messages sent during performing a DHT operation

In P2PSIP context the nodes can be divided into two classes. Some, or even all, of the nodes can form the DHT. These *service nodes* are denoted as "nodes" in the subsequent discussion. *Clients* are mobile devices that are not participating in the DHT. They are just utilizing the services of the DHT nodes by sending PUT/GET requests to the service nodes.

The contact information lookup time in call setup operations and the registration time are equal in Kademlia. In both cases a node lookup operation is performed. Thus, we need to focus on determining the lookup time (*lookup latency*).

In this study we use the standard Kademlia parameters for the P2PSIP DHT, thus the maximum number of nodes in the network is $N = 2^{160}$. Churn is modeled with a global constant p, which is the probability that a message is sent to an active contact that replies with an answer message. It can be determined if the probability distribution of the nodes' session times is available.

A. Lookup overhead

To calculate the message overhead, we need the average routing length (number of hops) l, as a function of the number of active nodes (n) in the DHT. In Kademlia, the routing tables of the nodes are divided into *logn* buckets. The buckets stores k neighbors in different distances to the node. This ensures that lookup operations have *O(logn)* complexity, which could lead to the conclusion that only one bit is improved in each routing step (after each step, the lookup result is one bit closer to the target). However, this is the worst-case scenario. As shown in [5], in reality, a contact selected in a routing step usually matches more than one bit; thus, the probability of improving at least δ bits is, if the bucket size is k:

$$F(\delta,k) = P[X \geq \delta] = 1-\left(1-\frac{1}{2^{\delta}}\right)^{k} \tag{1}$$

The probability of improving exactly δ bits is:

$$f(\delta,k) = P[X = \delta] = F(\delta,k) - F(\delta+1,k) \tag{2}$$

The average number of extra bits improved can be determined by calculating the expected value of (2). Thus, the total number of bits improved in a routing step:

$$t(k) = 1+\sum_{\delta=0}^{\infty}\delta\cdot f(\delta,k) \tag{3}$$

In each hop, t(k) is the average number of bits improved. Hence, the total number of hops (average routing length) needed to find the target:

$$l = \frac{\log_2 n}{t(k)} \tag{4}$$

By calculating the average message overhead C (number of messages sent during one lookup operation), we can estimate the load on the nodes in the system. [4] defines the formula for the message overhead in the DHT:

$$E[C] = kpl + \frac{kl}{1-(1-p)^{k}} \tag{5}$$

If the number of lookup operations performed during a given time is known, the number of messages processed by a single node can be determined by using the message overhead formula.

B. Message overhead per node

The total message overhead R is the number of messages received by a node per second. It is the sum of the client registration overhead R_{reg}, the call initiation related overhead R_{call} and the DHT maintenance overhead R_{mnt}.

$$R = R_{reg} + R_{call} + R_{mnt} \tag{6}$$

C. Registration overhead

Registration is the process of storing the contact information of a client in the DHT. In a Kademlia DHT, stored values must be updated (restored) periodically to prevent expiration. Since each update involves performing a lookup operation, the update frequency greatly affects the overall traffic in the network.

Let the update frequency be denoted by T_t and the average client online time by T_{online}. Clients perform a registration at the time they join the network. Therefore each client performs at least one registration while it is online. The average interval between registration operations (T_{reg}) is:

$$T_{reg} = \frac{T_{online}}{\left\lceil \frac{T_{online}}{T_t} \right\rceil} \tag{7}$$

A lookup operation, which generates $E[C]$ messages, is performed by c clients in every T_{reg} seconds; thus, the message overhead caused by client registrations per service node:

$$R_{reg} = \frac{cE[C]}{nT_{reg}} \tag{8}$$

D. Call initiation overhead

As it is explained in [1] the complete call establishment includes DHT lookup operation, SIP session establishment between a caller and a callee and NAT traversal. Here the call initiation related traffic is understood as the result of the contact information lookup operations performed when a call is set up. We estimate this by using a global calls per second parameter φ. If there are n nodes in the network, the number of clients is c and the message overhead of one lookup operation is $E[C]$, then the average number of messages sent to a DHT node in one second:

$$R_{call} = \frac{c \cdot \varphi \cdot E[C]}{n} \tag{9}$$

E. Maintenance overhead

To keep their routing tables up-to-date, Kademlia nodes perform lookup operations for the buckets whose contacts have not generated any traffic for a given period of time. If there are n nodes in the DHT, the number of non-empty buckets is $\log_2 n$. The non-empty buckets which have not received any updates for T_{rfs} seconds are refreshed. Since we know the number of messages received by a node in a given time, we can also estimate how many contacts are updated in each bucket. The total number of contacts updated in bucket j during T_{rfs} seconds is:

$$\frac{1}{2^j} \cdot T_{rfs} \cdot (R_{call} + R_{reg}) \tag{10}$$

To get the number of buckets that are updated we calculate the number of messages received by a node during T_{rfs}:

$$z = T_{rfs} \cdot (R_{call} + R_{reg}) \tag{11}$$

These z messages touch approximately $\log_2 z$ buckets. Thus, the number of inactive buckets that have not received any updates, and as a result are updated, is:

$$b = \max\left(0, \log_2 n - \lceil \log_2 z \rceil\right) \tag{12}$$

The DHT node performs a lookup operation for each inactive bucket, thus the maintenance overhead is:

$$R_{main} = \frac{E[C]b}{T_{rfs}} \tag{13}$$

III. Results and Discussion

In this section we apply the analytical model derived in the previous section to investigate a set of different scenarios. The global parameters used by the analysis are as follows. The average number of phone calls per day per user measured in the US is 9 [6], thus based on this, we set φ = 1/9600. Values stored in the DHT are refreshed in every hour, thus T_t = 3600. The online time for service nodes T_{online} = 19800 is based on the median supernode session time observed in Skype [7].

A. All nodes in DHT

The simplest approach is that all nodes that participate in the system also participate in the DHT (c=n). This is a straightforward approach because it does not require any decision making whether or not to participate in the DHT. The most significant benefit is that the load on each node is relatively low. However, as the size of the DHT grows, the maintenance traffic and the routing latency also increase. These two factors are driving to the opposite directions.

Since the DHT consists of mobile nodes, we assume a high churn. High churn comes from a number of factors. First, mobile nodes are sensitive to energy. Second, as the name implies, mobile nodes tend to move, and there could be frequent changes of IP address if the phone moves from one WLAN access point to another. Alternatively the connection is switched from WLAN to cellular when the phone leaves the hotspot. Churn was modeled with p=0.7 (30%), which means that almost every third messages is lost.

The plot in Figure 1 illustrates the dependency between the number of nodes in the system (n), and the message overhead per node (R). As it can be seen, the overhead increases with the number of nodes. This is due to increase of the average routing length and the message overhead of lookup operation. The larger the DHT, the more hops are needed to reach the target of the lookup. Nevertheless, the scalable architecture ensures that the message overhead is evenly distributed over the network. With one million nodes, around one message per second is received by each node. Using the average energy cost per message value 0.3 J that we measured in [2], one message per second per each node corresponds to 1080 J of energy per hour, which means roughly 4-5 hours of uptime with a fully charged Nokia N95 smartphone [8].

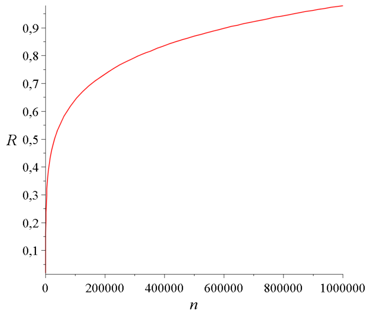

Figure 1. Message overhead per node (all nodes in the DHT)

B. Mobiles use DHT as service

Another strategy is that only a subset of the nodes in the system participates in the DHT and the rest are clients.

A complication of this strategy is how to elect the nodes to be part of DHT. From the mobile users point of view being elected to the DHT is bad. The extra DHT traffic consumes the resources, increases communication, and drains battery. So a key issue is to come up with a strategy that allows fair participation to the DHT.

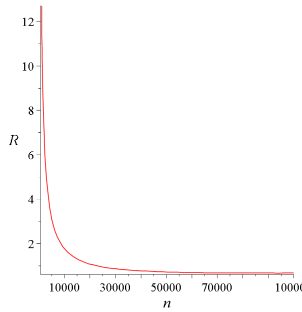

Figure 2.
Message overhead per node (R) as a function of the number of service nodes

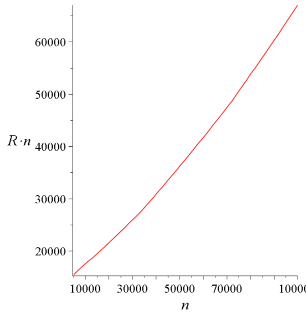

Figure 3.
Cummulative message overhead ($R \cdot n$)

Figure 2 shows a plot of the message overhead per node as the function of number of service nodes in the DHT serving a network with 1 million clients. It can be observed that having less than 10000 service nodes (1%) results in more than 2 messages per second, which is inacceptable in terms of energy consumption. Nevertheless, as the number of service nodes approaches the 10% limit, the overhead decreases below 1 message per second.

If we look at the energy consumption, it turns out that we have an interesting trade-off. To minimize the long-term cumulative energy consumption, we would like to have as little traffic as possible within the DHT. According to Figure 3, this can be achieved if the minimum number of nodes participates in the DHT. However, Figure 2 shows that the less nodes participate in the DHT, the higher the traffic and energy per node becomes, which can results in exceeded capacity.

It could be interesting to compare this strategy with the previous one from the battery consumption point of view. A possible modification is applying a strategy where by default everybody participates in the DHT but when the battery capacity drops below 50% the node leaves the network. This would create extra churn and thus extra traffic.

C. PC computers form DHT

In this strategy a subset of nodes forms the DHT. However, in contrast with the previous case, there is now a difference between the capabilities of the nodes. We assume that the DHT consists of powerful server computers. Thus the DHT is used to implement the server functionality in a distribution fashion. At the same time the distributed server could take advantage of the scalability and robustness aspects built-in in the DHT algorithms.

The typical case for this strategy is when the operator runs the DHT. The possibility to start with a small initial investment and add new servers to the DHT would allow the needed scalability. Currently deployed DHTs used in very large scale applications, such a serverless BitTorrent trackers, where over one million nodes participate [3], indicates that the solution has the potential to scale up to support a massive number of users.

Because the devices forming the DHT are server computers, we assume that the DHT is very stable. Churn is small and arises mainly from hardware faults and maintenance actions. We used p=0.99 (1%) for the plots.

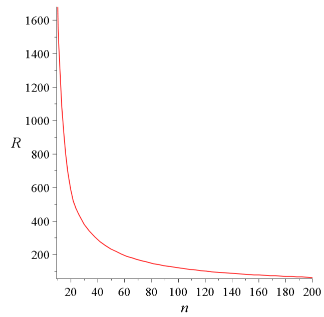

Figure 4. Message overhead per node (PCs form the DHT)

Figure 4 shows the number of messages received by a service nodes in a one million user DHT. Since PC-based servers can serve much higher number of requests than mobile devices, smaller number service nodes are required in the DHT.

D. Other observations

Table II. classifies the traffic experienced by DHT nodes by type. It shows the quantitative predications for a few typical configurations. The first column shows the baseline case where a single (virtual) server handles the complete traffic. The second column shows some sample numbers when we assume that the DHT consists of 50 nodes. The last column shows the data for the “everybody in the DHT” case, having all the 10 million clients function as service nodes as well.

As it can be seen the registration traffic is the main source of the load in the first two cases. Therefore it would be tempting to reduce the need for periodic re-registration.

Unfortunately, a longer re-registration period increases the probability that the DHT contains outdated data. To find the optimum point, we would require further understanding of the user behavior since the registration need arises when the user is no longer accessible at the IP address that was previously stored in the DHT.

The main difference in the third case is the increase of the maintenance overhead. While having 10 million service nodes greatly reduced the message overhead per for the call and registration traffic, more DHT nodes resulted in much higher maintenance traffic.

TABLE II. MESSAGE OVERHEAD PER SECOND PER SERVICE NODE IN DIFFERENT SIZE NETWORKS

Traffic type	10 million users, 1 service node	10 million users, 50 service nodes	10 million users / service nodes
Registration	41200	1050	0.02
Call	14000	360	0.007
Maintenance	~0	~0	1.4
Total	55200	1110	1.427

We also analyzed the case how traffic per node varies as a function of the percentage of nodes in DHT in the 10 million nodes case. Our results show that when the percentage increases at first, the traffic per node drops dramatically. However, after 3% there is hardly any effect. In fact after 6% the traffic per node starts to slightly increase due to the maintenance traffic. With this data suggests that the optimum percentage of nodes in DHT is around 1-2%. A larger DHT does not create any benefit but instead adds complexity and forces more and more nodes to be in the DHT. A better strategy would to rotate the nodes that participate in the DHT according to some fair mechanism. In the simplest case it could be a round robin approach and in the more complicated strategies we could take the nodes current energy level and participation history into account.

IV. RELATED WORK

The number of papers that investigate DHTs and P2PSIP from the mobile device point-of-view is very small. Matuszewski and Kokkonen [1] presented the first mobile P2PSIP implementation together with results of measurements. They, however, did not analyze the registration delay as well as call setup delay as the function of DHT network size.

Harjula at al. [9] discussed the challenges and problem areas in P2PSIP from the viewpoint of standardization and related research. They provided a glance to the existing and emerging solutions for P2PSIP.

Heikkinen at al. [10] applied analyzed P2PSIP from business perspective. Using a variant of Schoemaker's scenario planning they developed and analyzed most likely scenarios for mobile P2PSIP real-time services.

Our earlier work measured the energy consumption of DHT at mobile phones [2] and proposed backward compatible changes that allow major energy savings in mobile use of DHT [11].

V. CONCLUSIONS

In this paper we analyzed battery consumption and traffic overhead in P2PSIP mobile networks. Our analytical model and the result of battery measurements in the Mainline BitTorrent DHT network have allowed us to investigate different possible deployment scenarios for P2PSIP. Our analysis showed that in a one million user P2PSIP network, at least 30% of the mobile nodes should participate in operating the DHT to remain energy-efficient. However, if fixed servers are also used in operating the DHT and the DHT can handle load balancing by putting more load on fixed nodes than mobile nodes, the percentage of mobile service nodes can be lowered. Our analysis showed that the most dominant traffic overhead is related to registration traffic. This can be lowered by increasing the re-registration periods, but this can also results in increased lookup latency or lookup failures if the churn is high. Furthermore, although this analysis was based on Kademlia, future work could include investigating other DHT protocols for P2PSIP. Single hops DHTs, such as D1HT [12], have higher maintenance overhead but provide better latencies and lower lookup costs, which make them promising candidates.

REFERENCES

[1] M. Matuszewski, E. Kokkonen, “Mobile P2PSIP - Peer-to-Peer SIP Communication in Mobile Communities,” in Proc. of 5th IEEE Consumer Communications and Networking Conference (CCNC), 2008

[2] I. Kelényi, J. K. Nurminen: Energy Aspects of Peer Cooperation - Measurements with a Mobile DHT System. 43th IEEE International Conference on Communications (ICC 2008), Beijing

[3] J. Falkner. M. Piatek, J. P. John, A. Krishnamurthy, T. Anderson, “Profiling a Million User DHT,” in Proc. of Internet Measurement Conference 2007

[4] Di Wu, Ye Tian and Kam-Wing Ng . Analytical Study on Improving DHT Lookup Performance under Churn. In Proc. of the Sixth IEEE International Conference on Peer-to-Peer Computing, 2006

[5] D. Stutzbach, R. Rejaie, “Improving Lookup Performance over a Widely-deployed DHT,” in the Proc. of INFOCOM 2006, pp. 1-12.

[6] H. Verkasalo, A Cross-Country Comparison of Mobile Service and Handset Usage, Licentiate thesis, Helsinki Univ. of Technology, 2006

[7] S. Guha, N. Daswani, and R. Jain, "An experimental study of the skype peer-to-peer voip system." [Online]. Available: http://saikat.guha.cc/pub/iptps06-skype/

[8] J. K. Nurminen and J. Nöyränen, “Energy-Consumption in Mobile Peer-to-Peer – Quantitative Results from File Sharing,” in Proc. of Fifth IEEE Consumer Communications & Networking Conference (CCNC) 2008.

[9] Harjula E, Hautakorpi J, Beijar N & Ylianttila M. Peer-to-Peer SIP for Mobile Computing: Challenges and Solutions Seet B-C (ed.) Mobile Peer-to-Peer Computing for Next Generation Distributed Environments, Hershey, USA: IGI Publishing, 2009

[10] M. V. J. Heikkinen, M. Matuszewski, H. Hammainen, Scenario planning for emerging mobile services decision making: mobile Peer-to-Peer Session Initiation Protocol case study, International Journal of Information and Decision Sciences, Volume 1, No. 1 / 2008, pp. 26-43

[11] I. Kelényi, J. K. Nurminen: Optimizing Energy Consumption of Mobile Nodes in Heterogeneous Kademlia-based Distributed Hash Tables. 2nd IEEE NGMAST, 2008, Cardiff

[12] L. Monnerat and C. Amorim. D1HT: A Distributed One Hop Hash Table. In Proc of the 20th IEEE IPDPS, April 2006.