Future Network and MobileSummit 2010 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2010
ISBN: 978-1-905824-16-8

ROBUST: Reliable Overlay Based Utilisation of Services and Topology for Emergency MANETs

Grant P. Millar, Emmanouil A. Panaousis, Christos Politis
Wireless Multimedia & Networking (WMN) lab, Kingston University, London, UK
Tel: +442084172653, Email: {g.millar,e.panaousis,c.politis}@kingston.ac.uk

Abstract: Emergency services and networks are an emerging area of scientific research. During an emergency scenario such as forest fires, earthquakes, tsunamis and terrorist attacks rescuers need to establish communication to coordinate their actions by using smart and lightweight mobile devices. To this end, autonomous networks should be utilised to support the afore communications. Mobile Ad-hoc Networks (MANETs) are a characteristic paradigm of IP-based autonomous networks that can be deployed during critical emergency missions. We propose utilising a peer-to-peer (P2P) paradigm when designing application layer communication and data sharing technologies between participants of the MANET. The architecture therefore must allow nodes to send and retrieve data without knowledge of the complexities of the network. To this end we propose a Distributed Hash Table (DHT) architecture which we optimise for use in these situations and prove our theorem to be more efficient in such cases than its current counterpart.

Keywords: P2P, DHT, clustering, emergency, MANET.

1. Introduction

Mobile Ad Hoc Networks (MANETs) consist of a collection of devices (nodes) whom communicate in a completely decentralised manner. Nodes are actively able to send, receive and forward messages. In this paper we consider the notion of MANETs applied to emergency scenarios. Our work has been done within the context of FP7 ICT-SEC PEACE[1] project which investigates the provisioning of day-to-day emergency communications in next generation all-IP networks. We consider the term 'emergency scenarios' to envelope a plethora of extreme disaster situations such as but not limited to; forest fires, terrorist attacks, major floods and earthquakes. In these catastrophic events normal communication infrastructures have a high likelihood of being damaged beyond functional working conditions. Repairing such infrastructures can take a great amount of time, and emergency personnel (fire brigade, police, paramedics) need functional communications the second they arrive on scene in order to coordinate and put into action any disaster contingency plan they may have conceived as a forethought. In these stressful conditions reducing the setup time of the network can save countless lives. The instantaneous setup of MANETs makes them an ideal candidate for use by those first responders on scene in disaster recovery.

The lack of centralisation built within MANETs should adhere developers to apply the same peer-to-peer paradigm when building applications and services. For example when node A wishes to contact node B and does not know any other information except the fact the node is referred to as 'node B' it must utilise some distributed name lookup

[1]For more info visit: http://www.ict-peace.eu/.

scheme which should return an IP address from the name. The same concept can be applied in many services making the transition from traditional client-server networks to mobile ad hoc networks where the services depend on a system which would normally rely on a central entity. Examples of such services include; DNS, P2PSIP, Distributed File Systems (DFS), Instant Messaging (IM) and general information sharing using rich media such as images. All of these services can be combined to create a media rich P2P group collaboration environment.

In this paper we consider the requirements of users whom have a need to communicate and share data which would normally be carried out in a client-server fashion. In MANETs an obvious solution to such problems is introducing a decentralised database system. The specific system we propose as a solution to the aforementioned problem is referred to as Distributed Hash Tables (DHTs). DHTs are peer-to-peer overlay networks where nodes can lookup data and need not know its specific location in the network before hand. All that is required to complete a search request (which we refer to as a lookup here on in) is a single piece of meta data such as the name of a person, file name, or some other identifier.

The rest of this paper is organised as follows. In section II, we discuss related work done within the realm of DHTs. In section III, we describe our proposed methodology towards the provision of peer-to-peer services for MANETs in emergency cases. In section IV, we illustrate and discuss the results of our experimentation through simulation. Section V concludes this paper mentioning our plans for future work.

2. Related Work

A number of DHTs have been proposed by the research community for tackling the problem of the dissemination of distributed information over a network, and some have even been implemented in past research for primary use in homogenous networks akin to the internet with prominent architectures such as; Pastry [1], Chord [2], CAN [3], and Bamboo [4]. In this section we look specifically at what we feel are the most relevant DHTs designed for improving such architectures which can be applied in MANETs and the major differences when compared to homogenous DHTs.

The paper which proposes the DHT architecture Etka [5] explores the possibility of two opposite options of creating a DHT efficient architecture for MANETs. The results show a lower lookup delay and a higher Packet Delivery Ratio (PDR) when using a hybrid protocol. However due to the oversight of a lack of a single MANET routing standard, we will not consider this idea when proposing our architecture.

The MADPastry DHT [6] concept introduces the idea of a more proximity-aware DHT. This means that nodes no longer have completely random ID's akin to conventional DHTs built for large networks such as the internet. The approach laid out in the paper suggests to add a new type of node acclaimed a *landmark node* which sends out a one hop periodic beacon in the physical MANET. Nodes within close proximity to this node take on its node ID prefix, creating an approximate overlay topology to that of the physical network.

The paper [7] introduces the concept of improving the Bamboo DHT by using a hierarchical approach. However it does not go into detail about the architecture, but does provide results to show lower lookup latency, higher lookup success rate and lower overhead traffic. We believe that clustered architectures shows promise for improving

DHT performance and will use this as a paradigm for designing our architecture.

3. Proposed Methodology

In this section we give an overview of the main characteristics of the DHT which target critical functions to create a more efficient overlay when deployed in MANETs.

3.1 Bamboo as a basis

We choose to examine Bamboo because of its documented ability to handle a high frequency of nodes joining and leaving the overlay (churn) while maintaining a high lookup success rate compared to similar overlay approaches such as Chord as proven in [4]. This is very beneficial in MANETs in emergency situations because of the possibility of having a large number of obstacles which the physical layer radio waves may find it hard to communicate between. This consequently causes nodes to drop from the DHT and rejoin again rather frequently. In this paper we specifically target the overlay routing algorithm used in order to reduce firstly the number of logical hops, and translate this into fewer physical hops by creating a greater proximity aware overlay.

Within the research community a large area of DHT based research has appeared on the selection of neighbours in the DHT routing table, and how to create an efficient proximity-aware neighbour selection (PNS) algorithm. The DHT routing table has the effect of creating a more sturdy DHT by sampling nodes within close proximity to where we joined the network (coincidentally this is usually a node close to us) and by also creating routes to nodes whom are at the opposite side of the logical overlay. Bamboo employ's two PNS techniques. The first, Global sampling and the second sampling our neighbours' neighbours.

3.2 ROBUST DHT

We entitle our overlay approach as ***ROBUST DHT***, an acronym for Reliable Overlay Based Usage of Services and Topology. The fact that ROBUST is based on Bamboo gives it a solid foundation for handling a high level of churn suitable for situations in MANETs where the physical transmission can at times be intermittent.

The main concept of our architecture which differs from that of Bamboo is in both overlay routing and topology management in an effort to enforce less logical overlay hops, and at the same time translate that into less physical hops, whilst still being highly reliable. The aim of ROBUST is to limit Bamboo maintenance traffic while trying to decrease lookup speed. To achieve this we introduce a number of new concepts to the DHT as documented below.

The first most noticeable change from traditional DHTs like Bamboo is the introduction of a hierarchical system similar to that proposed in [7], our clustering technique divides up the N nodes in the DHT network into a number of $[C_1,\ C_2, ..., C_{\frac{N}{\log_2 N}}]$ clusters each with a size of S nodes. When nodes first join the DHT, they must contact a bootstrap node. In the proposed scheme the bootstrap node is a cluster node responsible for routing requests of a maximum number of S nodes in the DHT. Once the cluster node receives the join request it checks the number of nodes within its cluster, if the number of nodes within the cluster is less than S the node will join the aforementioned cluster, and said cluster node will be responsible for the nodes DHT routing outside of the cluster. If the number of nodes within the cluster is greater than S then the cluster node will forward the join request to its neighbouring cluster nodes, and the

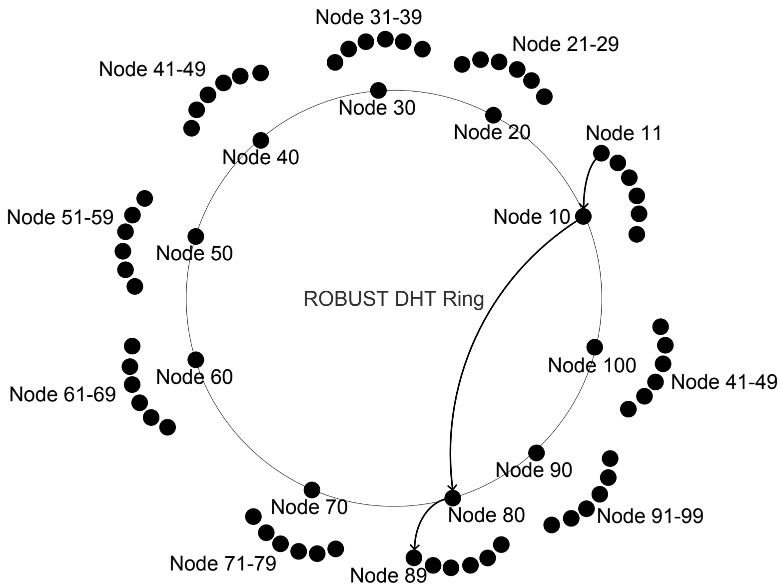

Figure 1: An example of how routing works in ROBUST from node 11 to node 89

same process will happen again until the node is allocated within a cluster. Therefore the complexity of the number of hops a join request should traverse before a cluster is found for any given node can be denoted as $O(\log_2 C)$. Once a neighbour has established its cluster node it will then take on the node ID $p \cdot \lfloor(\frac{d}{10^x})\rfloor$ where p is the cluster nodes ID prefix and $x = \lfloor \log p \rfloor + 1$, d is the nodes original node ID. The value of N is calculated periodically by the cluster nodes passing information about the number of nodes within their cluster in a round robin type fashion initiated at the cluster node with the lowest node ID, once the number is computed with the total arriving back at the cluster node with the lowest node ID, this will then be broadcast to all of the other cluster nodes. Figure 1 depicts an example of a clustered ROBUST DHT ring, showing the routing process from node 11 to node 89.

Using such a structure in our architecture means that we no longer need the Proximity Neighbour Selection (PNS) as used in Bamboo. The reason for this is due to the fact that we no longer need inter-cluster nodes to store such information. With nodes joining the overlay in a proximity aware scheme these tables would only serve the purpose of lowering routing hops for which the complexity in Bamboo is $O(\log_2 N)$. However we prove that this can be reduced by routing information first to the cluster node, then linking all cluster nodes within the DHT using a Plaxton [8] routing scheme. In this case, to resolve a lookup a node will first forward the lookup request to its designated cluster node, this node will then forward the request in no more than $O(\log_2 C)$ hops to the cluster node responsible for the node which contains the data. The algorithm for the amount of clusters needed in the DHT is denoted as:

$$C = [\frac{N}{\log_2 N}] \tag{1}$$

We partition the network in a way that each cluster has a size of $S = \log_2 N$ nodes. We assume that in each cluster any node is no more than one logical hop away from the marker node. Thus routing complexity from the cluster node to any node inside the cluster it is responsible for is $O(1)$.

So for end to end communication in the overlay level we have the following complexity for the DHT routing algorithm:

$$O(1) + O(\log_2 C) + O(1) \tag{2}$$

From (1), (2) we will have a complete complexity of $O(1) + O(\log_2 \frac{N}{\log_2 N}) + O(1) = O(\log_2 \frac{N}{\log_2 N}) = O(\log_2 N - \log_2(\log_2 N))$.

In Bamboo, the DHT routing complexity is equal to $O(\log_2 N)$. Therefore to establish that the proposed routing scheme has less complexity we have to compare $O(\log_2 N - \log_2(\log_2 N))$ and $O(\log_2 N)$. If we prove that $\log_2 N - \log_2(\log_2 N) \leq \log_2 N$ we have proven that our algorithm has lower complexity. Obviously this is proved if $\log_2(\log_2 N) \geq 0$. Considering that the minimum number of nodes in a cluster is 2 we can show the following: $\log_2 N \geq 1 \rightarrow \log_2(\log_2 N) \geq 0 \rightarrow \log_2 N - \log_2(\log_2 N) \leq \log_2 N \rightarrow O(\log_2 N - \log_2(\log_2 N)) \leq O(\log_2 N)$. The consequence of having an algorithm with lower complexity is that of lower lookup time and less overhead, which should also equate to more successful lookups due to timeouts not being exceeded.

In order to address scalability more clusters can be added to the overlay. To achieve this cluster nodes maintain a cluster node routing table of five cluster nodes above its ID and five cluster nodes below its ID. Initial cluster nodes find their counterparts using a cluster node lookup which is routed in the same manner as a Bamboo DHT lookup. Once initial cluster nodes have been established they share information about the number of nodes they each contain by utilising the data storage facilities in the DHT which is updated periodically. Initially in the case where $\frac{N}{\log_2 N} > 2C$ the cluster nodes will split their ID space by two, at every point $[p_1,\ p_2, ..., p_i]$ in the ID space as denoted by $p = \frac{h}{C}$ where h is the maximum hash ID and C is equal to the number of clusters in the DHT before the next round of splitting. The splitting starts with the cluster node with the second lowest ID prefix p_2 as the prefix cannot be less than one, therefore a new cluster node will be designated with the node ID p'_i where; $p'_i = p_{i-1} + \frac{p_i - p_{i-1}}{2}$. Once $\frac{N}{\log_2 N} > 2C'$ for the new number of clusters the next iteration will begin again from the start of the ID space. We can see that in an evenly distributed ID space, when we split the cluster in two, we will move nodes with a complexity of $O(\log_2 N)$ per original cluster so for moving all of the needed nodes in the DHT to new clusters we have a total complexity of $O(\frac{(\log_2 N)^2}{\log_2(\log_2 N)})$. In the case where there are less nodes in the network compared to the number of clusters needed we can create a threshold of $\frac{N}{\log_2 N} < 2C$ peers, so that only if a whole iteration of clusters are not needed we start to scale down the number of clusters in order to keep the ID space equally distributed. When scaling down the number of cluster nodes we remove all cluster nodes in the sequence of $[p_2,\ p_4, ..., p_j]$ where $j = 2k$, and $2 \leq j < i$ and $k \in \mathbb{Z}^+$. This adaptive quality means that no one cluster node will be swamped with too many inter-cluster nodes, alleviating bottlenecks. Our decision to split the overlay in this manner as apposed to adding individual clusters to the overlay ID space when needed is to keep the ID space symmetrical and balance the load.

In MANETs we expect nodes to be mobile, therefore when a node moves from one position to another it may actually be closer to another cluster node. To address this formality we use a function called proximity synchronisation. Cluster nodes send out a beacon periodically to each of the nodes within their cluster, these nodes then forward the beacon to any nodes around them (with a 1 hop TTL) who are not part of the same cluster. The result is that if a node moves closer to another cluster node, it can ping the newly discovered cluster node and compare the latency with its current cluster node. If the latency of the new cluster node is lower, then the node sends a cluster join request to the relevant cluster node, if the cluster is not full, the node leaves the overlay with his current node ID, and rejoins with a new node ID in correlation to the new clusters prefix. Mobility has further impacts on the DHT such as partitioning where two islets of nodes form due to a weak bottleneck in the network, currently nodes would continue probing the down node IP until the connection is re-established, however further solutions will be proposed allowing dynamic merging of partitioned overlays in future work.

To optimise the overlay further we also propose a cluster node election scheme. In this scheme current cluster nodes periodically poll all of the nodes within the cluster for a reliability metric $[u_1,\ u_2, ..., u_{\log N}]$ every 300 seconds so that we do not flood the network with cluster nodes changes. The reliability metric is calculated using the following equation; $u_i = w_1 t_i + w_2 b_i + w_3 l_i$. To allow flexibility in each scenario w_i values are weights used to increase or decrease the impact factor of a given metric. The individual values $[t_1,\ t_2, ..., t_{\log N}]$, $[b_1,\ b_2, ..., b_{\log N}]$, and $[l_1,\ l_2, ..., l_{\log N}]$ are equal to a threshold of fuzzy logic denoted as $[v_1,\ v_2, ..., v_j]$ where a higher value means higher reliability. The threshold v_j is based on predetermined parameters, but for each case the data values used to calculate the reliability are as follows; maximum throughput denoted as t_i in bits per second, remaining battery power b_i as a percentage of power remaining and latency denoted as $l_i = (\sum_{m=1}^{s} l_m)/s$ where s is the number of nodes in the cluster, and l_m denotes the latency between the node and each other node within the cluster. Combining these three metrics to give us a reliability metric allows us to simply select which node is best suited to be the cluster node based on which node has the highest reliability metric. If two or more nodes have the same highest metric value, the node with the best data value chosen as either t_i, b_i or l_i can be selected. When a new node is selected as the cluster node, the previous node first swaps ID's with the new cluster node, then sends a cluster update message to all nodes in the cluster with the new nodes information. As cluster nodes pull each others leafsets in iterations, the new cluster node notifies itself to its neighbour cluster nodes and the subsequent convergence of the new cluster node is no more than $O(\log_2 C)$ phases.

4. Performance Evaluation

We ran preliminary simulations in the network simulator NS-2, implementing a basic functionality of the above described algorithm. In our NS-2 implementation nodes join the overlay, a number of data keys are added to the overlay (an equal amount to the number of nodes) and lookups are generated at random with a frequency of 10 lookups per second, giving a total of 6000 as we have simulated 10 minutes of real time. This paper lays out the foundation of the DHT architecture, and as such presents only preliminary results, the authors wish to point out that factors such as churn and topology control overhead will be investigated in future work.

One of the main factors of a peer-to-peer overlay network is the overhead maintenance traffic needed to ensure nodes routing tables are kept up-to-date, Figure 2 shows us that for ROBUST the overhead traffic is equal to or less than Bamboo in all situa-

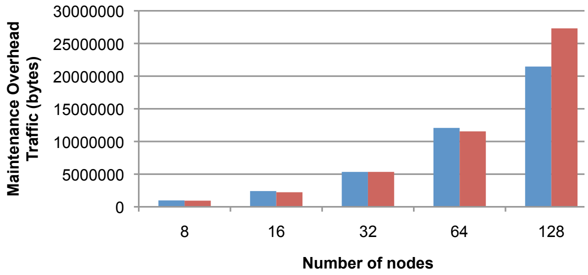

Figure 2: The maintenance overhead traffic.

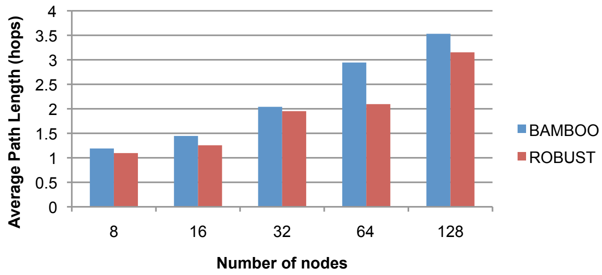

Figure 3: The average path length.

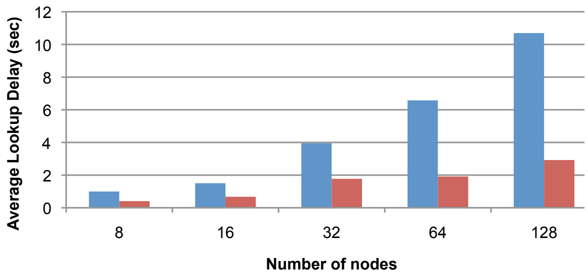

Figure 4: The average lookup delay.

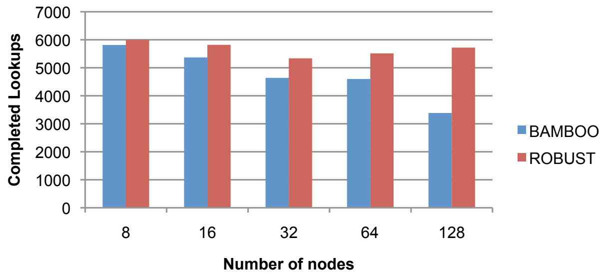

Figure 5: The number of completed lookups.

tions bar 128 nodes. The reason the overhead is higher for ROBUST in this case is due to the fact that ROBUST resolved far more lookups than Bamboo, logically requiring more overhead. As we would expect, due to the proximity-aware architecture of ROBUST the average path length shown in Figure 3 is lower for all number of simulated nodes. The results show that overall in terms of successful lookups ROBUST outperforms Bamboo as we can see from Figure 5 that no matter the amount of nodes, more lookups are successful in ROBUST. In Figure 4 we observe that while the Bamboo lookup delay increases significantly due to overall higher delay, the cluster node selection based on reliability keeps low latency nodes at the core of the network, resulting in faster lookup times even with many nodes.

While these results are promising and show that ROBUST can outperform Bamboo in MANET situations where high latency is inevitable, we believe once the entire ROBUST algorithm is implemented in NS-2 the results will be even more dramatic, this is something we will endeavour to investigate in future work.

5. Conclusions

In this paper we have proposed a novel solution to optimise the performance of DHTs in mobile networks. To achieve optimisation we have proposed a clustering algorithm to create equal sized clusters of nodes brought together in a proximity aware fashion. Routing in our DHT under normal operation is always passed from the node getting/putting data, directly to its responsible cluster node, it is then forwarded to the cluster node responsible for the node containing the data key, and then to the target node in one hop with a lower complexity than that of Bamboo. We have proposed scaling methods to ensure that the DHT will not have bottle necks due to clusters being too large. To further prove the basis of our theory preliminary results from simulations show that the algorithm performs well compared to Bamboo in environments where latency between nodes can be high.

Our plans for future work include further investigation and optimisation of ROBUST performance with respect to error resilience, scalability and reliability issues. Furthermore, security considerations and implementations to protect ROBUST will be designed. To this end, secure DHT-based routing to guarantee correct message delivery, secure node ID assignment, secure data storage and trustworthiness among the different peers will be addressed. Security mechanisms will be based on an initial key management scheme which will support symmetric cryptographic algorithms and message authentication codes to provide authentication, integrity and confidentiality within the P2P network.

6. Acknowledgment

This work was undertaken in the context of the project ICT-SEC-2007 PEACE (IP-based Emergency Applications and serviCes for nExt generation networks) with contract number 225654. The project has received research funding from the European 7^{th} Framework Programme.

References

[1] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Proc. of the IFIP/ACM Middleware 2001*, pp. 329–350, 2001.

[2] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. on Networking*, vol. 11, no. 1, pp. 17–32, Feb 2003.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," *Proc. of ACM SIGCOMM 2005*, pp. 161–172, 2001.

[4] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "Opendht: a public dht service and its uses," *Proc. of ACM SIGCOMM 2005*, pp. 73–84, 2005.

[5] P. H., D. S.M., and H. Y.C., "Ekta: an efficient dht substrate for distributed applications in mobile ad hoc networks," *Proc. of IEEE Workshop, WMCSA 2004*, pp. 163–173, 2004.

[6] T. Zahn and J. Schiller, "MADPastry: A DHT Substrate for Practicably Sized MANETs," *Proc. of ASWN 2005*, June 2005.

[7] Z. Tian, X. Wen, Y. Sun, W. Zheng, and Y. Cheng, "Improved bamboo algorithm based on hierarchical network model," *Proc. of ISECS CCCM 2009*, pp. 297–300, 2009.

[8] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," *Proc. of ACM SPAA 1997*, pp. 311–320, 1997.

