Decentralized, Resource-Aware Information Management and Delay Tolerant Networks in Command-and-Control

Markus Brückner, Liz Ribe-Baumann
{markus.brueckner, liz.ribe-baumann}@tu-ilmenau.de

Abstract: A robust, decentralized information management system and reliable transport of information between disconnected agents is essential to the success of command and control activities in disaster relief scenarios. We present work-in-progress aimed at developing a distributed hash table (DHT) protocol and a delay-tolerant network that meet the specific requirements of disaster relief scenarios through, among other things, the integration of location and resource awareness.

1 Introduction

During the response and recovery phases after a disaster, large amounts of data are produced and must be stored and transfered to the relief workforces in order to ensure timely, informed, and well coordinated relief efforts. The success of relief efforts depends not only on the right people being sent to the right place, at the right time, with the right equipment, but also their access to up-to-date information pertinent to their tasks and their ability to save and share important information that they are gathering. Such information may include data about environmental conditions, locations of resources, progress of relief efforts, positioning and expertise of relief workforces, or lists of tasks. Unfortunately, the storage and transfer of data in a widespread disaster scenario is complicated by several factors: The amount of information and number of users in the system increases incredibly fast (even disaster in a confined space like the WTC attacks spur a high amount of communication [K$^+$04]); much of the system operates on limited battery power; landlines and base stations may be damaged; and relief workforces must be sent to areas without means for communicating directly with the command center.

The work-in-progress presented in this paper considers two approaches aimed at ensuring reliable storage and transfer of data for the control and command of disaster relief efforts where each node knows its geographic position. As we can not anticipate all possible uses of this work, we focus on one specific type of data generated in a disaster: operation and technical logs which are maintained at every field unit (see [fKuK99] for details). These logs contain information about nearly every event a unit encounters as well as reactions to it. As it is typically used for documentation purposes only, we do not consider it to be time critical. On the other hand, we argue that providing this data to non-data-generating units in a timely manner might prove beneficial both as a backup, improving data availability, and for data mining, improving situational awareness. Transporting the information drawn from this heightened situational awareness back into the field in an automatic fashion would

lessen the burden on both the communication backbone as well as the operators in the control centers.

To facilitate such applications we present partial solution ideas for the development of a highly scalable, location and resource aware distributed data management system (with resources such as power, bandwidth, etc.) as well as a data transport system which enables the forwarding of the data through poorly connected areas. The rest of the paper is organized as follows: We present the data management system in Section 2, which is based on a distributed hash table (DHT) and maps data to network nodes and provides routing algorithms for lookups (see for example [SMK+01]). In Section 3, we present the data transport system which uses a delay tolerant network (DTN) [Fal03] that, in order to minimize delay and maximize delivery probability, exploits prior knowledge about node movements (in our use case, couriers). Section 4 provides a brief conclusion and an outlook to future work.

2 Data Management - Resource Aware Overlay

We consider a DHT to be resource aware when it spreads load among nodes such that node failures or timeouts due to the (over)use of specific resources are minimized, and assume that each query routed over a node uses a portion of that node's available resources. Our overlay construction is motivated by the small-world graphs explored by Watts and Stogatz [WS98], the construction of the virtual network coordinates Vivaldi [DCKM04], and the small-world DHT Symphony [MBR03]. We integrate resource levels into nodes' coordinates and then choose links based on nodes' coordinates' distances in the hopes of reaching a "small-world-like", resource aware DHT similar to Chord [SMK+01].

We assume that each node x possesses two dimensional geographic coordinates $x^g = (x_1, x_2)$ and a resource level $x_r \in \{1, \ldots, r_{max}\}$ for some network-wide maximal resource level r_{max}. Each node also possesses an additional "height" dimension, similar to Vivaldi network coordinates [DCKM04]. Vivaldi coordinates emulate a mass-spring system to estimate nodes' locations in a self-organized manner and use an additional height dimension to distance single nodes from the *entire* network. For the special case of 3-dimensional Vivaldi coordinates, two nodes x and y with coordinates $x^c = (x_1, x_2, x_h)$ and $y^c = (y_1, y_2, y_h)$, have Vivaldi-distance:

$$d_V(x^c, y^c) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} + x_h + y_h.$$

Ledlie et al. [LGS07] showed that this additional height dimension, which was designed to accommodate the expensive last hop in routing that separates a node from the highly connected internet core, is integral to accurate latency estimates. We use a height dimension to express each node's resource level and distance nodes with low resource availability from the entire network. Thus, a node x with resource level x_r receives resource height $x_h = (r_{max} - x_r)/r_{max}$. Essentially, the shorter the distance between two nodes, the more likely they will be linked, and thus, nodes with lower resource levels have fewer incoming neighbors to incur routing load.

HandleSearchMessage($x^c, d, NodeList$):
1. if ($|d_V(x^c, thisNode^c) - d| \leq \varepsilon$)
   then sendSearchMessageReply()
// if thisNode at correct dist. to x, reply to x
2. else
   addToNodeList($thisNode, NodeList$)
   y = findClosestNeighbor($x^c, d, NodeList$)
   sendSearchMessage($y, x^c, d, NodeList$)
// find $y \notin NodeList$ in routing tables with
// smallest $|d_V(x^c, y^c) - d|$, forward to y

Figure 1: Pseudo-code for $thisNode$'s handling of a coordinate link search message.

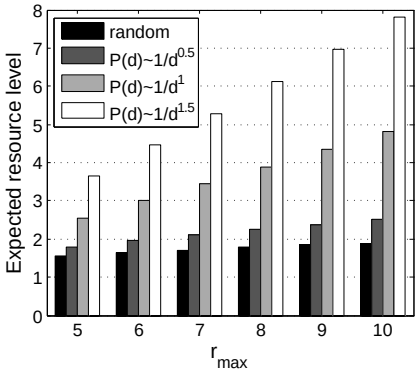

Figure 2: Expected resource level of random coordinate link for various r_{max}'s and probability distributions over distances.

Overlay Initialization. Our DHT functions similar to other ring shaped DHTs (e.g. Chord [SMK+01]), with keys in the interval $[0, 1)$ which wraps around, typical join and failure protocols, and unidirectional routing in the key space. To join the DHT, a node x chooses an identifier $NodeID_x$ uniformly at random from the keyspace and contacts some participating node y with $NodeID_x$. Then y performs a key-lookup to find the node responsible for $NodeID_x$ and x positions itself on the ring of keys, establishes links to its k nearest neighbors on the ring, and assumes responsibility for the range of keys preceding $NodeID_x$ as defined in Chord. Next, x establishes links for two kinds of routing tables which are used jointly for routing: location aware *finger links* based on nodes' NodeIDs as in DHash++ [DCKM04] and *coordinate links* chosen based solely on their distance to x in the network coordinate space.

Coordinate Links. Coordinate links are established based on a discrete probability distribution over the distances between nodes in the network coordinate space: the smaller the distance between two nodes' network coordinates, the higher the probability that they are joined by a coordinate link. To find a new coordinate link, a node x draws a *distance* d from its probability distribution and then initiates a search for a node y at approximately distance d from x in the network coordinate space:

$$d_V(x^c, y^c) \in (d - \varepsilon, d + \varepsilon),$$

for some tolerance level $\varepsilon > 0$. Thus, x sends a search message to a random neighbor with its coordinates x^c and distance d, which is forwarded at most some fixed number of iterations (see Figure 1 for more detail). If a suitable node is found, it is added to x's coordinate-link routing table, otherwise a new distance is drawn and the process restarts. Since nodes have dynamic coordinates, links' coordinates must be monitored and a link must be renewed if its distance has changed significantly.

A node x's probability distribution over the distances depends on x_{rad}, the radius of the coordinate space as measured from x's position (details are omitted here), and the maximal

resource level r_{max}. Distances are chosen as increments of $1/r_{max}$ for factors from 1 to $x_{rad} \cdot r_{max}$. For example, the probability distribution for which the probability for choosing d is inversely proportional to d is:

$$P_x(d = k/r_{max}) = \frac{1/k}{\sum_{i=1}^{x_{rad} r_{max}} 1/i}, \quad k \in \{1, 2, \ldots x_{rad} r_{max}\}. \tag{1}$$

In order to examine the effectiveness of the coordinate-links or the entire system, many assumptions must be made about the distribution of resources and nodes. As an example, Figure 2 shows the expected resource level of a single coordinate link for Zipf-distributed resources - where high resource availability is rare - and uniformly distributed nodes within a cone-shaped area with radius 100. In this case we see that for the distribution in (1) the resource level for a random link is substantially higher than for $P_x(d) \sim d$ (labeled 'random'). However, such observations depend strongly on the assumptions of node and resource distributions.

3 Data Transfer

We propose that the minimization of transport overhead and delay could be reached by optimal adaption of the routing system to nodes' current movement patterns (i.e. passing messages to nodes that will move towards the target as soon as possible). As our solution is intended for relief workers where we expect a certain amount of pre-known missions (e.g. couriers approaching field units at known locations). We intend to use the mission data as a base for routing optimization. This work draws ideas from the work of Lindgren et.al. [LDS03] who proposed using repetitive contact patterns of nodes to optimize message forwarding. We were also inspired by geographic forwarding algorithms as in [Fin87] which rely on the geographic position of nodes to transport data.

System structure. In order to keep routing flexible and extendable, we minimize the interfaces between the routing algorithm and the rest of the system. As the routing optimization is based on the knowledge of node trajectories, we expect a per-node information store to provide access to this information. The necessary data should be exchanged whenever two nodes meet in order to provide system-wide dissemination. Each node provides its own trajectory data based on information from the mission control or – in the case of human-controlled vehicles – route guidance. Our proposal would be to derive the necessary information from mission descriptions given electronically. This is based on the assumption that at least the operation controllers should know where field units are located and are therefore able to provide rather accurate descriptions of the target of a mission. By keeping the information about node movement abstract and separate from the actual source of this data we facilitate later system extensions with other sources of trajectory data like movement prediction algorithms which provide limited data even on systems without pre-known mission goals (e.g. units in search-and-rescue missions).

Movement-based forwarding. Routing – especially geographic routing – is essentially a well-understood problem for which a set of solutions exists. As with the data storage, we use these solutions and adapt them to the problem at hand. Looking at nodes' trajectories in a common space immediately displays a similarity to a two-dimensional map. While this similarity is clear, it can be misleading: trajectories may intersect but routing of messages along them is only possible if the corresponding nodes meet. In order to capture this we have to extend the model by one dimension: the time.

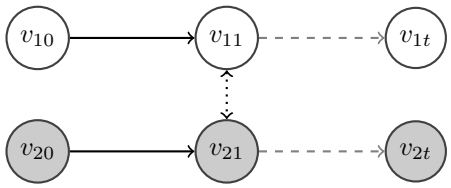

Figure 3: A connectivity graph of two nodes meeting once

By introducing the time into the picture we define the concept of a *temporal map*. The movement of the node is now a function of time: $T : \mathbb{R} \to \mathbb{R}^2$ where T is the trajectory of a node. By calculating the distances of all trajectories within the prediction window, we are able to derive a connectivity graph that models the possible paths of messages through the system by adding virtual intermediate nodes to the graph for every communication opportunity. An example for such a graph is given in Figure 3. The white and gray sets of vertices each represent one node at different points in time. The black solid edges represent messages traveling inside a node's buffer for some time, while the dotted edge represents a communication opportunity. The gray, dashed edges are mere helpers to simplify future route calculation. Their edge weights are set to a neutral element with regard to the cost function for the path calculation.

With the right weights assigned to the edges, optimizing the forwarding of a packet from node n_1 to n_2 is modeled by finding the minimum-cost path through the connectivity graph from v_{10} to v_{2t}, a problem that is well understood and solved e.g. by Dijkstra's algorithm. Whenever a node has to take a forwarding decision (i.e. whenever the opportunity to communicate arises) it can calculate the shortest path to the target and check whether the crossover edge representing the current communication attempt is contained in that path. If it is, the message is forwarded to the communication partner, otherwise it is kept in the internal buffer. Depending on the selection of the cost function, the path could be optimized for minimum delay or maximum delivery probability.

4 Conclusion - Future Work

We presented solution ideas to the problem of storing and transporting non-time-critical data in a disaster. Our solution is divided into a DHT-based storage subsystem providing reliability despite widespread failures and limited resource availability and a DTN-based

data transport system adapting to existing movement patterns. While our use cases are currently aimed at an electronic version of operation and technical logs, we can not fully anticipate future uses of the system. We expect that the widespread availability of the data will increase situational awareness and prevent mistakes due to incomplete or outdated information, but this must still be investigated. Future work should focus on improving our understanding of the scalability and robustness of the system. On the one hand, replication schemes for the data management system need to be developed in order to maintain a desired level of availability given nodes' locations and resources. On the other hand, data prioritization and reliability levels should be integrated into the transport system to provide further optimizations of message transport. Furthermore, the potential for the optimization of message transport using the remaining infrastructure after widespread failures should be assessed. However, since this system is dissimilar to existing communication infrastructures, we are currently not pursuing the direct integration with existing networks.

References

[DCKM04] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. *SIGCOMM '04*, pages 15–26, 2004.

[Fal03] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03*, pages 27–34. ACM Press, 2003.

[Fin87] G.G. Finn. Routing and Addressing Problems in Large Metropolitan-Scale Internetworks. Technical report, University of Southern California, Marina del Rey, Information Sciences Institute, 1987.

[fKuK99] Ständige Konferenz für Katastrophenfürsorge und Katastrophenschutz. Führung und Leitung im Einsatz. Download: http://www.katastrophenvorsorge.de/pub/publications/DV100-SKK.pdf, 12 1999.

[K+04] Thomas H. Kean et al. The 9/11 Commision Report. Technical report, National Commission on Terrorist Attacks upon the United States, 6 2004.

[LDS03] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic Routing in Intermittently Connected Networks. In *SIGMOBILE Mobile Computing and Communication Review*, volume 7, 7 2003.

[LGS07] Jonathan Ledlie, Paul Gardner, and Margo Seltzer. Network coordinates in the wild. In *In Proceedings of USENIX NSDI'07*, 2007.

[MBR03] Gurmeet S. Manku, Mayank Bawa, and Prabhakar Raghavan. Symphony: Distributed Hashing in a Small World. In *In Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, pages 127–140, 2003.

[SMK+01] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*, pages 149–160, 2001.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of'small-world'networks. *Nature*, 393(6684):409–10, 1998.