Infocosm: Towards Collective Decision Making in Mobile Social Networks

Divyakant Agrawal
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5110, USA
agrawal@cs.ucsb.edu

ABSTRACT

As Internet based services and mobile computing become ubiquitous, society becomes increasingly reliant on these communication media to accomplish critical and time-sensitive tasks such as information dissemination and collaborative decision-making. The dependence on these media magnifies the damage caused from their disruption, whether by malicious intent or natural disaster. For instance, in the event of a natural disaster, such as the earthquake in Haiti or the tsunami in Japan, disruption of the centralized mobile and Internet infrastructures impedes information spread and often leads to chaos, both amongst the victims as well the aid providers. Decentralized and ad-hoc mechanisms for information dissemination and decision-making are paramount to help restore order. We propose Infocosm, a mobile social network that utilizes direct device communication to enable group decision-making, or consensus, without reliance on global communication services. Infocosm focuses on minimizing the system resources, to prolong the lifetime of the power constrained devices, by minimizing communication overhead, computational complexity, and persistent storage size. Infocosm provides a simple and intuitive system to enable large-scale coordination amongst non-expert users. Due to the mobility of the users in Infocosm, limited range of point-to-point communication, and the ad-hoc nature of the infrastructure, all the participants in the system cannot communicate with each other. Estimating the number of participants (or a global count) itself becomes a challenge and hence, traditional notions of consensus or quorum based protocols for agreement cannot be used. We, therefore, rely of threshold and time limit based approaches to reach an agreement. In this presentation, we will explore various heuristics and models to estimate group participation to aid users in reconciling consensus.

1. MOTIVATION

From Tahrir Square to Wall Street, new technologies, such as social networks and mobile computing devices, are enabling people to quickly organize in a decentralized manner. Social networks are unintentionally serving as groupware to synchronize and facilitate human interactions [4]. The phenomenon of information diffusion and influence in social networks has been the interest of recent research and modeling. In an abstract sense, all popular social networks enable a user to express an idea and subsequently propagate the idea through a network of peers. This straightforward diffusion of information and the simple interface has enabled people to organize in a lightweight manner which is essential to facilitate large scale group interaction. However, this approach is not entirely decentralized, as these tools rely on Internet services to act as a centralized coordinator for user messages throughout a network. In the event of a natural disaster or an administration turning adversarial to curb a movement, Internet access may become limited or unavailable. Several recent events demonstrate this scenario. With the 2010 earthquake in Haiti, rescue works relied on text messaging (SMS) to coordinate efforts, as cell phone networks strained under failure and overloading [6, 7]. Planned protests at the Bay Area Rapid Transit (BART) in San Francisco, USA resulted in cellular service being cut in order to stave off protests [1]. Lastly, and perhaps most infamously, Hosni Mubarak's government shut down Internet access to Egypt in an attempt to thwart not only social network coordination, but also privacy filters, such as Tor, that bypass censorship firewalls [3].

Despite the availability of a centralized service, certain actions still require coordination of a large group, whether it be a protest of conditions or organizing humanitarian efforts. Without a single point of communication, planned actions can become disjoint and unclear, and result in a reduced effectiveness. We define a problem in the described environment as how can a system disseminate an idea, or proposal, amongst a network of peers in order to ascertain user's intention and determine an expected outcome. As knowledge and context of the proposal is required to determine success, a proposer should specify a tipping point (*quorum value*), which is the number of users that need to agree in order to achieve a consensus. In this context, we use consensus as the consent of a specified group size for a proposed value, and not the stringent definition of consensus where all non-faulty processes agree on a single value. Due to the lack of a central authority and due to the mobility of the users, the number of users that will observe a given ballot is unknown. As a result, classical notions of quorums, group size estimation, or consensus in a static distributed system [8, 12] cannot be applied directly in this problem setting. Since a consistent view of the number of users that will observe a proposal is not known, an application specific quorum value threshold is needed to approximate consensus. Quorum value is application specific since the number of people required to organize a meeting at a database conference is very different compared to the number required to effectively organize a city protest.

The group decision making problem framed in a disconnected environment becomes technically challenging. Users that are mobile, are likely to have different views on the state of the ballot due to the lack of a single point of truth and observing intentions of disjoint peers. Divergent views need to be reconciled between users, so an approximate view can be consistently determined in order to have similar state when a proposal expires. The proposals may be critical for users, so the state of the proposal should be persisted beyond volatile memory. *The reconciliation of disconnected replicas,* *efficient persistent storage, managing concurrent proposals, and an understanding of consensus call for a database system solution to this timely problem.*

The reliance on a centralized cellular or Internet access does not prevent many modern mobile devices from communicating with its peers at a large scale. Many smartphones have capabilities to directly communicate with other mobile devices within a limited range, including IEEE 802.11, Bluetooth, or Ultra Wide Band. These communication media provide the ability to discover peers within a few hundred feet allowing for the construction of a mobile peer-to-peer (**P2P**) network to exchange information [12]. Leveraging these networks and motivated by the need for decentralized organizational tools, we introduce Infocosm, a mobile P2P database that enables group decision making without relying on centralized services. While mobile P2P databases for disaster situations or military applications have been proposed before, specifics were not provided. Moreover, our focus is on group decision making which is beyond communication overlays [11]. The name Infocosm symbolizes our goal to create an aggregate global view of information composed of smaller pieces of information that can be difficult to piece together and individually cannot represent the global picture. In this presentation we will highlight the need for Infocosm and the research challenges associated with enabling group decision making in a P2P mobile environment. We will also present a brief overview of a research prototype that is being developed at UC Santa Barbara to highlight some of the implementation challenges in developing mobile social network technologies.

2. PRELIMINARIES

Infocosm is a database system running on mobile devices that communicates in a P2P fashion with other mobile devices in its vicinity. In Infocosm, a user can propose a question (the **proposal**), optionally accompanied by a suggested answer (the **value**). The proposal is broadcast to all users within the proposer's direct communication range. Since the communication range is limited and the users are mobile, a proposal might eventually be relayed to areas where the original user initiating the proposal is not present. We use the term **proposer** for the user that introduces the ballot to a set of peers who have not received the proposal earlier. The proposer can therefore be the original initiator or a relay node. The mobile agents forward this proposal, allowing users to agree with the proposal, suggest a new value, or reject the proposal. New values are only allowed if none of the proposed valued have reached the tipping point. The users that respond non-negatively to a proposal are **subscribers** to the proposal. Every proposal has an associated **expiration time** after which it is no longer valid; a proposal is spread until its expiration time is reached. The proposal is encapsulated within a **ballot** that contains the proposal, the suggested value, the expiration time, the proposer's unique ID, the minimum number of users to achieve a consensus, and the set of users who have accepted each of the ballot's potential values. All subscribers are notified of the proposal's **outcome** either when a value reaches the tipping point or when the proposal expired. The outcome of a proposal is the values which received votes above a predetermined threshold. Since Infocosm is designed to operate in disconnected modes where all users are mobile and a centralized single point of truth for the ballot may not exist where data is incomplete, the vote counts are associated with error bounds.

Figure 1 describes a sample ballot for the proposal. The ballot is expressed in JSON format for the ease of exposition; Infocosm stores and transmits a ballot in a compact compressed binary format. In this example the proposer suggests that at least twenty participants are required to achieve the tipping point. The

```
ballot{
 proposal: "Gather to demand face-to-face
           PC meetings for SIGMOD",
 expiration: "2012-05-24 10:00:00",
 suggestedValue: "Canyon Room 5/24 11:00",
 quorumValue: 20,
 userAccepts: {
   "Canyon Room 5/24 11:00" : [31083,
      13091, 38919, 900941, 109381],
   "Mesa Hall 5/24 14:00" : [13134]
 }
 key: 107074168843,
 proposerId: 31480
}
```

Figure 1: A sample ballot to demand conference review changes.

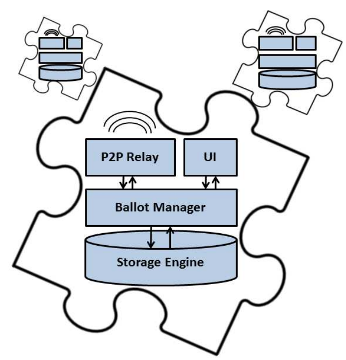

Figure 2: Internal Infocosm Components.

snapshot of Figure 1 captures a scenario where an alternative value has also been proposed. The ballot lists both the proposed values: `Canyon Room at 11 am` that has six acceptors and `Mesa Hall 5/24 14:00` has a single vote. For brevity, the error values and divergence in versions is not shown, but Infocosm tracks it internally. The ballot is broadcast to all peers within range. Each peer with an active Infocosm instance will notify its user of a new ballot. The user can take action on the ballot by either *accepting* the ballot, *relaying* it, *proposing* a new value, or *ignoring* the ballot. The proposer for this ballot is notified of this user's intention. An *accept* signifies that the peer agrees with the proposal, and will relay the proposal to all future peers. A *relay* signifies that the peer will not commit to the proposed value but will subscribe to the ballot and rebroadcast to future peers. If the peer disagrees with the suggested value of the ballot, an alternate value can be proposed. *Ignore* states that the user rejects the ballot, and that Infocosm should ignore all future messages regarding this ballot. The proposer adds all accepted users and proposed values to the ballot, makes the ballot locally persistent in Infocosm, and rebroadcasts the updated ballot to all users in range. On receipt of a message, a peer determines if the ballot contains new information or should be ignored. Additionally, a peer does not rebroadcast a ballot with an identical state (i.e. no new acceptors or values) to peers already aware of the ballot.

3. SYSTEM DESIGN

Infocosm is composed of four major components shown in Figure 2. The *storage engine* provides persistence for the observed ballots and the actions taken by the user. The *P2P relay* notifies the ballot manager of new peers observed, incoming ballots, and ballot proposal responses. The relay also manages proposal broadcasts and responses from the ballot manager. The *UI* module interfaces the ballot manager with the user to acquire responses about proposals and notify with updates on the subscribed ballots. The *ballot manager* coordinates all components, decides what information to relay between components, ignore and broadcast ballot information to new peers, and inform the user about the status and outcome of a ballot.

Several challenges arise in designing an efficient ballot manager. Battery life, network bandwidth, and storage capacities are limited and many optimizations are needed to minimize Infocosm's footprint and resource consumption. An instance of Infocosm on each node stores detailed information about a ballot so that the ballot manager can appropriately orchestrate coordination between the components. However, since network communication is expensive, peers only exchange a compressed ballot header. This compressed header should be sufficient to determine if additional information is needed to synchronize the ballot views of two nodes.

The frequency at which ballots are transmitted must also be optimized. While a great deal of literature exists on gossip protocols and the properties of epidemic communication in mobile environments, additional context can be leveraged in communication protocols [12]. This context can include the expiration time, the mobility patterns, and popularity of a proposal. Mobility patterns will be especially important when a small subset of the peers are mobile, while majority remain static and are less likely to interact with new peers.

Due to the decentralized nature, timestamps alone cannot accurately determine if peers have a consistent view of a ballot. As users move and are disconnected, their versions of the ballot might diverge. As a result, Infocosm must address important research challenges in reconciliation of the divergent ballots, and their respective counts, eliminating duplicates from the counts, as well as tracking the lineage of the ballot versions as they diverge. For instance, a user U_j receives the ballot from another user U_i. After accepting the proposal, U_j moves to the vicinity of U_k and passes the ballot. Now, if U_k moves into the vicinity of U_i, U_i must be able to eliminate the duplicate counts through the lineage of divergent versions. Storing and exchanging the full accepted user set to track lineage is expensive in terms of bandwidth and computation. A naïve optimization involves comparing a hash an ordered set of users (such as bloom filters); however, this approach incurs a high cost to merge large sets and is also an approximate set membership. Additionally, storing and exchanging large sorted sets can be too costly for storage and computation on a device where resources are finite. We are currently exploring using a combination of approaches, using probabilistic data structures, such as bloom filters, to determine set membership quickly, or utilize multi-sets for quick merges and rely on sampling or sketch based techniques to estimate of set membership [5, 9, 13]. Alternatively, Infocosm can leverage de-duplication techniques for comparing large sets and quickly identifying differences in the data set [2]. Lastly, if a large set of peers have frequent interactions, such as conference attendees, a coordinated checkpoint can also be constructed.

Most importantly, Infocosm should also be able to reason about the confidence of consensus beyond ballots reaching a designated threshold of votes. However, due to the uncertain nature of the network topology and communication, true consensus cannot be achieved due to the impossibility of agreement and validity. Often traditional distributed systems notions, like consistency, assume a static number of nodes. Many of these assumptions were reexamined with a rise in popularity of dynamic systems, where the number of nodes varies over time. Research into *Group Size Estimation* and dynamic system modeling will guide efforts into building robust models of consensus for a mobile environment [8, 10]. Since the mobile agents differ from dynamic systems with spatial and temporal patterns, Infocosm considers empirical observations about the mobility of a user and the churn and mobility of peers when modeling consistency. Finally, a feedback mechanism on the outcome is requested from the proposer, in order to reinforce models that accurately stated which proposed value achieved consensus.

In addition to the above mentioned challenges, Infocosm must also address multiple issues, such as privacy of the users, potentially intermittent centralized sources of truth, malicious behavior, and trustworthiness. Discussion of all these challenges and possible approaches to overcome these challenges will be the subject of our presentation at the workshop.

4. REFERENCES

[1] Washington Post: BART San Francisco cut cell services to avert protest. `http://wapo.st/tMomtM`, 2011.

[2] D. Bhagwat, K. Eshghi, D. D. E. Long, and M. Lillibridge. Extreme binning: Scalable, parallel deduplication for chunk-based file backup. In *MASCOTS'09*, pages 1–9, 2009.

[3] The Telegraph : How Egypt shut down the internet. `http://tgr.ph/vidllq`, 2011.

[4] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some issues and experiences. *Commun. ACM*, pages 39–58, 1991.

[5] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31:182–209, September 1985.

[6] IEEE Spectrum : Why Haiti's Cellphone Networks Failed. `http://bit.ly/tnud7v`, 2010.

[7] Reuters : Cell phones and radios help save lives after Haiti earthquake. `http://reut.rs/vU1gMh`, 2010.

[8] D. Kostoulas, D. Psaltoulis, I. Gupta, K. P. Birman, and A. J. Demers. Active and passive techniques for group size estimation in large-scale and dynamic distributed systems. *J. Syst. Softw.*, pages 1639–1658, October 2007.

[9] A. Metwally, D. Agrawal, and A. El Abbadi. Why go logarithmic if we can go linear?: Towards effective distinct counting of search traffic. In *EDBT*, pages 618–629, 2008.

[10] A. Mostéfaoui, M. Raynal, C. Travers, S. Patterson, D. Agrawal, and A. El Abbadi. From static distributed systems to dynamic systems. In *SRDS*, pages 109–118, 2005.

[11] S. Shekhar and H. Xiong, editors. *Encyclopedia of GIS*. Springer, 2008.

[12] O. Wolfson, B. Xu, H. Yin, and H. Cao. Search-and-discover in mobile p2p network databases. In *ICDCS*, pages 65–, 2006.

[13] K. Wu, Y. Xiao, J. Li, and B. Sun. A distributed algorithm for finding global icebergs with linked counting bloom filters. In *ICC'08*, pages 2757–2761, 2008.