INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

SHARING OF CONTENT IN MOBILE ADHOC NETWORK

SNEHAL H. KUCHE, SONALI A. KARALE

Assistant. Professor, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati, India.

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

Mobile ad hoc network,

Network coding,

Peer-to-peer,

Haggle

Corresponding Author

Ms. Snehal H. Kuche

Abstract

Mobile devices are today capable of capturing various forms of multimedia and able to communicate wirelessly using increasing numbers of communication techniques. The owners and creators of local content are motivated to share this content in ever increasing volume; the conclusion has been that social networks sites are seeing a revolution in the sharing of information between communities of people. Mobile peer-to-peer systems have recently got in the limelight of the research community that is striving to build efficient and effective mobile content addressable networks. Along this line of research, a new peer-to-peer (P2P) file sharing protocol suited to mobile ad hoc networks (MANET). The main ingredients of this protocol are network coding and mobility assisted data propagation, i.e., single-hop communication. Network coding in combination with single hop communication allows P2P file sharing systems in MANET [2] to operate in a more efficient manner and helps the systems to deal with typical MANET issues such as dynamic topology and intermittent connectivity as well as various other issues that have been disregarded in previous MANET P2P researches such as addressing, node/user density, non-cooperativeness, and unreliable channel.

1. INTRODUCTION

In fact, nowadays, the market proposes a variety of mobile devices offering user-friendly interfaces, long-life battery autonomy, sufficient computational power and efficient wireless connectivity. This tremendous advancement triggers the necessity of supporting the very fashionable desktop applications in such mobile environments. Indeed, thanks to the efficient wireless connectivity offered by mobile devices (PDAs, smartphones...), users are frequently brought to locate and share content of interest (data, photos, videos, etc) with other members of the same spontaneous community.

Most peer-to-peer (P2P) file sharing systems (e.g., Gnutella, BitTorrent) are developed targeting wired IP networks and thus hardly work as intended in mobile ad hoc networks (MANETs) without modification[2]. Recently, several P2P schemes targeting MANET, MANET-optimized version of existing P2P schemes as well as clean-slate designs, have been proposed.

First problem is caused by the fact that the wireless channel is error prone. If a protocol is designed without considering potential errors, the performance of the protocol in real deployment will be seriously degraded. For example, TCP connections usually die out in multihop networks with lossy channel but most P2P protocols simply assume that TCP offers reasonable bandwidth. Secondly, number of users and user density should also be considered. In a file sharing scenario, the total number of users can scale up to tens of thousands of nodes, and theoretically, all of the nodes can be users running P2P protocols.

Mobile devices, such as smartphones and tablets, have increased the opportunities to access and share content while on the move [1]. Yet, most sharing schemes rely on intermediary services, such as Facebook and Twitter (for sharing status updates, pictures, and videos), and infrastructure for the underlying communication. The increasing dependence on online services puts strain on wireless access networks (e.g., 3G), forcing providers offload traffic onto local networks (networks (e.g., WiFi)which may also be overloaded or inaccessible. Complementary ad-hoc approaches to content sharing can over attractive alternatives to cloud services and can work together with infrastructure or independent of it, allowing users to share content without having to worry about overrunning their monthly data caps. Additionally, such sharing may open up new possibilities for social networking, file-sharing, games, and other applications that take advantage of user proximity and social interactions. P2P file sharing solutions are good candidates for such infrastructureless networks as they are based on multisourcing which balances resources consumption among peers and reduces the dependency on any central entity.

2 .LITURATURE REVIEW

Haggle leverages search to disseminate content based on its relative relevance to interested parties, by matching locally stored content against the interests of users. Search enables dissemination of content in order of how strongly users desire it, offering delay and resource savings by prioritizing the content that matters. Haggle, a system for ad-hoc content sharing, focusing on how search can be leveraged for relevance-aware dissemination. Unlike previous mobile content sharing systems,[3]Haggle prioritizes content based on how relevant it is to users, and allows highly desirable content to propagate faster than less desirable content even when the group of interested users is small (i.e., the content is not so popular, but those that desire it have strong interests). While other systems optimize for high delivery ratios and low aver- age delay for all content irrespective of its relevance to user

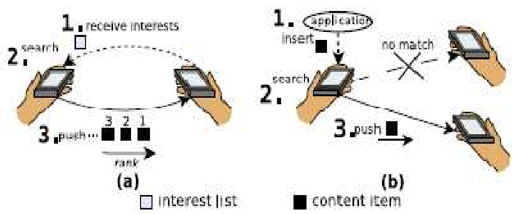


Fig.1 Content sharing in rank order, triggered by receiving new interests (a) or adding new content (b).

As shown in Figure 1 (a), two devices in contact share their interests (as detailed later, interests are actually content items that all nodes are interested in, obviating the need for a separate interest-sharing mechanism). The interests query the data store for matching content items, which are then pushed back in order of their search rank down to a congurable limit (compare this to how search engines only return the top 10 hits in a search). Received interests are stored and can further propagate to third-party nodes that repeat this search, pushing back content via delegate nodes. Similarly, new content items may also be pushed to neighbors when added by applications (Figure 1 (b)). Haggle can discourage the dissemination of low-relevance content in order to save battery, storage,bandwidth, and to reduce the spreading of “junk".

Push-based search dissemination as a primary mechanism for content sharing is one of the main contributions of Haggle when compared to related systems, such as DTN, PodNet and traditional pub/- sub systems. Another contribution is a new way to use forwarding algorithms to compute content delegates, i.e., nodes that carry content although they may have no interest in it. Comparable content-centric systems that to know of only disseminate among interested nodes. The lessons learned from designing Haggle's architecture, and then implementing it, comprise third contribution; figuring out the right abstractions and primitives and how they fit together into a system is a significant research undertaking. The architecture and prototype represent several years of research and experimentation that provide valuable insights into the most important aspects of building a real system [4]. Finally, privacy and security are crucial to any architecture. With ad-hoc content sharing, new strategies for securing content rather than hosts are needed. Nodes rely on each other for dissemination, and must therefore ensure that content can be shared securely without risk of flooding nodes with junk" or having malicious nodes disrupt the network. Schemes for learning of trusted content sources need to be devised, as well as mechanisms to set up trusted and secure groups based on social interaction and physical proximity.

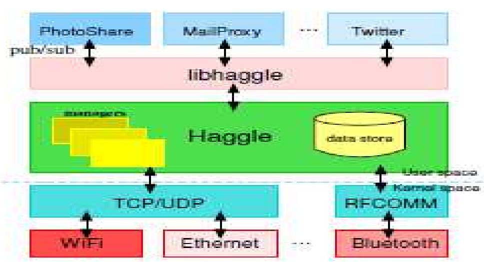
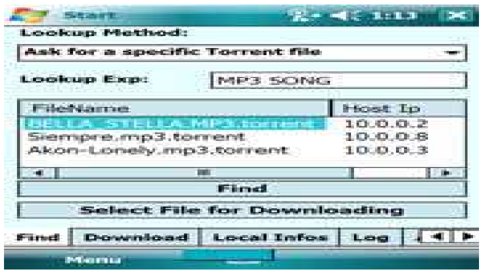


Fig.2 Haggles Implementation

1. CONTENT SHARING

Before starting a new sharing session, the user can choose between two versions of BitTorrent algorithms: The classical version and version adapted to mobile ad-hoc networks. Moreover, the BitHoc client overs a Wizard allowing the user to configure the parameters of BitTorrent (communication ports, choking slot duration, min/max number of peers, etc). Once the torrent file is obtained the BitHoc client can start the sharing session where it can either play the role of a leecher or a seed. It contacts periodically the local BitHoc tracker to get the current list of members of the same content sharing session (torrent). Using this list and the routing table, it manages the connections with the interested peers. Briefly a client implementing our algorithms exchanges pieces with close peers and only seeds distribute pieces across the network. Note that allow the user to pause or resume the download while conserving the session context. can also monitor in real time the status of the session (downloaded bytes, uploaded bytes, numbers of leechers, number of seeders, elapsed time, etc). Furthermore, the BitHoc client keeps in a log ¯le statistics on the content sharing session and provides different levels of event traces. It also manages the storage of the downloaded contents and their classification.



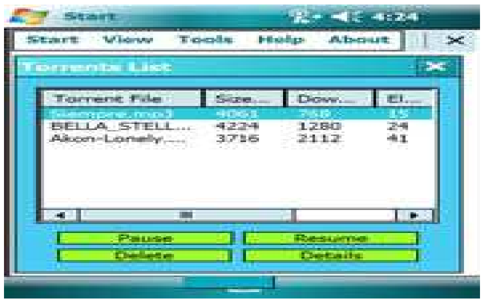


Fig. 3 Search Engine **Fig.4 BitHoc Client**

2. BITHOC ARCHITECTURE

Figure5 depicts the principal components of the BitHoc architecture and the interactions between them.

4.1 Content Publishing and Discovery

A user willing to share some content with the members of his community needs to indicate to the BitHoc client the location of the content in the mobile device file system. First the client creates a meta-info file (torrent file) that identifies in a unique manner a sharing session of the content. After that, the user publish (locally) the new torrent file and a short text description of the related content using the BitHoc search engine service, which will update the local torrent files database maintained in the underlying BitHoc tracker via HTTP messages.

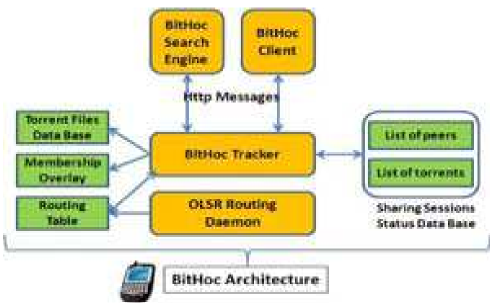


Fig. 5: BitHoc Architecture

A remote user, willing to share the same content, has to use the BitHoc search engine to find and download the torrent file. He specifies for that the name of the content or some key words related to its description. The request is sent via HTTP messages to its local tracker which looks for the closest match in its local database [5]. If there are no matches, it forwards the HTTP request to the other trackers in the discovery overlay. Then it presents the received results through an ergonomic user interface (see Figure 4). Based on the details of received answers (fitness to the search, number of peers involved in the sharing session, number of seeders, and number of leechers...), the user can choose the torrent file to download, then start the content sharing using the BitHoc client.

3. CONCLUSION

Haggle a system that allows mobile phone users to share content based on its relevance, allowing prioritization of content when node contacts are time limited. Our evaluation shows that search with relative rankings provides more exibility than the simple on/off matching of comparable systems, while content delegation proves useful when interest groups are partitioned and can be bridged by non-members. Haggle's source code is available online7 through an open source license, and provides a readily available platform for researchers to build and study mobile applications for ad-hoc content sharing. A wide variety of applications have already been built on top of Haggle, and we expect more to be implemented in the future. Many challenges remain, such as how to design efficient aging schemes, leverage infrastructure, and further enhance privacy and security. We also want to investigate inter-community dissemination and study other weighting and ranking schemes for search resolution.

REFRENCE

1. Bjurefors, F., Gunningberg, Nordstr om, E., and Rohner,C. Interest dissemination in a searchable data-centric network.In European Wireless (2010).

2. D. N. da Hora, D. F. Macedo, L. B. Oliveira, I. G. Siqueira, A. A. F. Loureiro, J. M. Nogueira, and G. Pujolle. Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks. Comput. Commun., 32(13-14):1445–1459, Aug. 2009.

3. A.-H. Leung and Y.-K. Kwok. On topology control of wireless peer-to-peer file sharing networks: energy efficiency, fairness and incentive. In World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a,pages 318 – 323, june 2005.

4. T. Repantis and V. Kalogeraki. Data dissemination in mobile peer-to-peer networks. In Proceedings of the 6th international conference on Mobile data management, MDM ’05, pages 211–219, New York,NY, USA, 2005. ACM.

5. M.K.Sbai, C.Barakat, J.Choi, A. Hamra, andT.Turletti. Adapting BitTorrent to wireless ad-hoc networks, in Ad-Hoc Now, Sophia Antipolis.