BitHoc: A content sharing application for Wireless Ad hoc Networks

Amir KRIFA, Mohamed Karim SBAI, Chadi BARAKAT, Thierry TURLETTI
EPI Planète, INRIA, France
Emails: {akrifa, mksbai, cbarakat, turletti}@sophia.inria.fr

***Abstract*—As a result of the wide spread of new efficient mobile communication devices, users are more and more willing to find on these devices (PDAs, Smartphones, etc) applications similar to the ones they are using on their desktop computers. We go one step in this direction and propose in this demo a software we have designed and implemented to enable content sharing among spontaneous communities of mobile users using wireless multihop connections. Our contribution consists in designing, implementing and testing a trackerless BitTorrent like application adapted to wireless ad-hoc networks. The proposed architecture includes two principal components: a membership management service and a content sharing service. As classical tracker-based BitTorrent membership management and peer discovery are unfeasible in ad hoc networks, we design the membership management service as a distributed tracker overlay that connects peers involved in the same sharing session. Using the membership information provided by the tracker overlay, the content sharing service schedules the data transfer connections among the session members by leveraging the multihop routing feature of wireless ad-hoc networks. Our test-bed composed of PDAs and smartphones equipped with WIFI adapters and Windows Mobile 6 operating system is used to validate the feasibility of our application and to evaluate its performance.**

I. Introduction

Thanks to the recent technological developments, ubiquitous mobile computing and communication have become a reality. In fact, 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. Among them comes the building of user communities that evolve in real time and that adapt to the dynamism of the network. One main objective of a community is the sharing of contents (data, photos, videos, etc) among devices having the same interest. In such a challenging environment, the burden of data sharing must be fairly distributed among all participants and since there are no dedicated nodes, one cannot rely on a centralized approach. Hence, P2P file sharing solutions are good candidates for such infrastructureless networks as they are based on multi-sourcing which balances resources consumption among peers and reduces the dependency on any central entity. But unfortunately, the P2P content sharing applications developed for the Internet can not directly be plugged and used into mobile devices [2]. Indeed, on one hand, their algorithms are not adapted to the constraints of multi-hop wireless networks. For example it is known that in a resource constrained environment, the choice of the peers to whom connect cannot be done independently of information on the underlying dynamic topology. Moreover, centralized peer management approaches like the tracker one in BitTorrent cannot be used either as the tracker can be far from and even invisible by some of the peers. On the other hand, due to resource constraints, Software Development Kits (SDK) proposed for mobile devices are only a subset of classical SDK(s) used for desktops which leads to incompatibility problems.

In this demo proposal, we present our open-source software solution for content sharing in mobile wireless ad hoc networks. We propose a BitTorrent [1] variant called BitHoc,adapted to the wireless ad hoc network environment, which is the result of our research work [2]. Our architecture consists of two main components: a membership management service (BitHoc Tracker) and a content sharing service (BitHoc client). BitHoc Tracker agents installed in nodes connect to each other in order to form the global distributed membership tracking service. To ensure content sharing, the BitHoc client decides in a distributed manner, using routing information, of the structure of the data exchange overlay (who exchanges with whom). It also manages the scheduling of pieces of data among devices. This service adapts the peer neighbor and piece selection strategies of BitTorrent to account for the topology of the network and the scarcity and shared nature of resources. The algorithms of BitHoc are described in [2].
We developed our software on mobile devices having Windows Mobile 6 as an operating system and equipped with WIFI adapters. As permanent topology information is required of BitHoc, we choose to use the proactive routing protocol OLSR [4]. For the evaluation, we have measured the performance of BitHoc in regards to the download time metric. The test-bed we used contains 14 PDAs and smartphones. This test-bed allowed us to experiment with the different features of our solution and to compare our version of the BitTorrent's algorithms to the ordinary Internet one in the wireless ad hoc environment. The performance analysis based on the experimental shows that BitHoc outperforms the classical Internet version.

The remainder of this paper is organized as follows. Section

This work has been supported by the ITEA European project on experience sharing in mobile communities (Expeshare).

II analyses the requirements of BitHoc. Section III describes its architecture. Section IV presents the demo and the test-bed we have used to test and evaluate our solution. Section V summarizes our contribution.

II. Requirements analysis

The objective behind our solution for P2P content sharing in wireless ad hoc networks is to cope with the problems related to the Internet version of BitTorrent in three regards:

- In the Internet version of BitTorrent [1], peers contact periodically a central rendezvous point called *Tracker* to obtain fresh information about the peers interested in a specific content. This membership information is dynamic since peers can join or leave the content sharing overlay (called torrent) at any time during the session. Since there are no dedicated nodes in a wireless ad hoc network and due to the big overhead engendered by the client/server architecture, it is important to introduce a distributed trackerless solution to manage the membership of the content sharing session. The BitHoc Tracker component of our architecture is designed for this purpose. Note that a distributed tracker architecture resists by nature to network partitioning whereas a centralized one does not.
- The Internet version of BitTorrent [1] supposes that the cost of sending data packets to peers is independent of their locations. In an ad hoc network, performance metrics like achievable throughput, delay, and energy consumption depend strongly on the number of hops to the peer node. So, it is clearly suboptimal and even unrealistic to deal with peers without considering the underlying topology. Furthermore, when applying the classical BitTorrent's incentives in a wireless multi-hop network, nodes fail to reciprocate data fairly among them. The content dissemination scheme is close to a wave transferring data from the initial seed to the farthest peers. Through new peer neighbor selection and content pieces selection strategies, our solution is topology-aware and ensures fair sharing. These strategies have been described in details in our research work [2]. Globally, we found that the best to do is to share the most data with physical neighbors to reduce the network overhead and to realize the maximum throughput, while sending pieces of content from time to time to far peers to provide them with material to share and keep them busy.
- Due to the incompatibility existing between desktop BitTorrent clients and the mobile environment, we started from the core open source BitTorrent library BTpeer [3] proposed for desktop computers and we rehabilitated its design and implementation to suit the windows mobile 6 environment using the appropriate SDK.

III. Design of our architecture

In this section, we present the architecture of BitHoc and its functionalities. Figure 1 depicts the principal components and the interactions between them. In the following paragraphs, we describe a typical usage scenario of our application in order to explain the features of the services and how they work together.

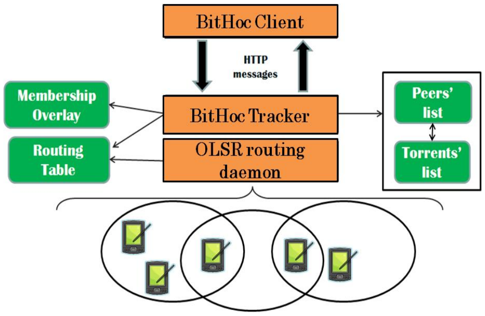


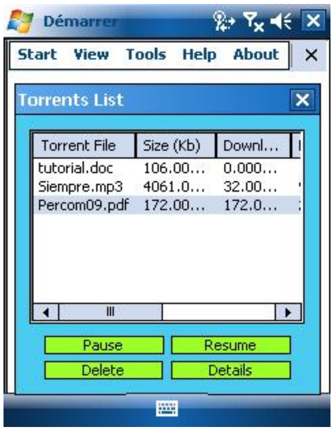
Fig. 1. Global architecture of BitHoc

When a peer wants to join or leave the sharing session, the BitHoc client informs the BitHoc Tracker about this event using a specific HTTP message. This local agent disseminates this modification to the other BitHoc Tracker agents in order to update their knowledge about the global membership information.The communications between Tracker agents are established in an event-driven fashion and use HTTP messages. Each tracker holds a HTTP server accepting HTTP requests from other agents and from the local BitTorrent client. The BitHoc Tracker component receives from the routing daemon up-to-date routing entries. In our testbed, the dynamics of the ad hoc network are captured by the OLSR routing protocol [4]. Each time the number of hops toward a given peer changes, the routing daemon fires an event, which will be caught by the BitHoc tracker and forwarded internally to the BitHoc client. This way we are sure the peer selection algorithm always uses the updated number of hops to other peers. The parameters of the communications among tracker agents like HTTP listening ports and IP addresses can easily be configured by users via an ergonomic GUI. In addition to these functionalities, the BitHoc Tracker allows the user to monitor in real-time the status of the overlay (shared contents, members of a session, current topology of the ad hoc network). He can even decide to keep traces about all the events in a file. For this, he just need to activate the tracing option provided by the application.

A user willing to share some content with the members of his community needs to point the BitHoc client to the location of the content in the mobile device file system. This component creates a meta-info file (Torrent file) that identifies in a unique manner a sharing session of the content. Before starting a new sharing session, the user chooses the version of BitTorrent algorithms to use. Two versions are available: The classical version [1] and our version adapted to mobile ad hoc networks [2]. Moreover, the BitHoc client offers a Wizard allowing the user to configure the parameters of BitTorrent (communication ports, choking slot duration, min/max number of peers, etc). By providing the Torrent file to the BitHoc client[1], a user can start either seeding or leeching the related content. The

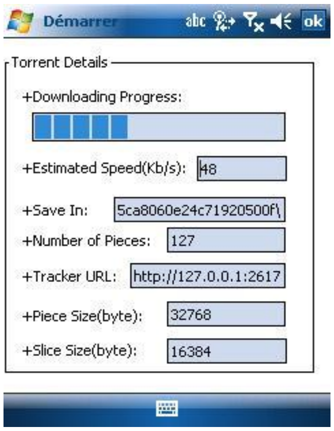
[1] The Torrent file distribution and retrieval are out of the scope of BitHoc. They can be done like in the Internet by other outband communication channels.

BitHoc client contacts periodically the BitHoc Tracker in order to get the current list of members of the content sharing session and the current status of the underlying topology. Using this information, it manages the connections with the interested peers. During the sharing, the user can pause or resume the download while conserving the session context. He can also monitor in real time the status of the session (Downloaded Bytes, Uploaded Bytes, Numbers of leechers, Number of seeds, Elapsed time, etc). Figure 2 and Figure 3 show screenshots of the monitoring graphic user interface. Furthermore, the BitHoc client keeps in a log file statistics on the content sharing session and provides different levels of event traces. The BitHoc client also manages the storage of the downloaded contents and their classification.



Torrent File	Size (Kb)	Downl...
tutorial.doc	106.00...	0.000...
Siempre.mp3	4061.0...	32.00...
Percom09.pdf	172.00...	172.0...

Fig. 2. List of Torrents



+Downloading Progress:

+Estimated Speed(Kb/s): 48

+Save In: 5ca8060e24c71920500f\

+Number of Pieces: 127

+Tracker URL: http://127.0.0.1:2617

+Piece Size(byte): 32768

+Slice Size(byte): 16384

Fig. 3. Torrent details

IV. Experimentation and Demonstration

A. Test-bed description

The wireless ad hoc network deployment that we used to test and experiment our solution for content sharing consists of 14 mobile devices including 7 PDAs and 7 smartphones. Each handheld is equipped with an IEEE802.11b wireless card. The characteristics of the two types of devices are summarized in Table I. The ad hoc connectivity is maintained thanks to OLSR daemons run by the different devices.

In our experiments, we constructed several network topologies containing a maximum of 8 hops. The objective of the realized swarm is to download a 4 MB MP-3 content. All PDAs were supposed to implement BitHoc and to participate to the sharing of the file. The original seed of the content was chosen randomly among the set of the 14 PDAs.During the demo session, we propose to show and run the same scenario using a subset of these PDAs.

	PDA	Smartphone
Name	HP iPAQ 214	HP iPAQ 614c
Processor speed	624 MHz	520 MHz
RAM	128 MB	128 MB
Operating system	Windows Mobile 6	Windows Mobile 6

TABLE I
Characteristics of the handhelds

B. Experimentation results

The metric tracked during our experiments was the download time of the different peers, which is one of the most important metrics from the user's viewpoint. Two versions of BitTorrent were tested over the settings described earlier in this section. The first version is the classical Internet version of BitTorrent, and the second one is BitHoc.

Figure 4 plots the average download time per peer as a function of the number of hops between the peer and the initial seed. Each point is computed by averaging the download time over several experiments. The figure shows that the classical BitTorrent engenders two to three times larger download times than BitHoc. Indeed, the classical BitTorrent leads to important network routing overhead (due to topology unawareness) and so fails to use efficiently the network capacity. Another important factor to consider when deploying BitTorrent in wireless ad hoc networks is the *data exchange reciprocity*, which provides incentives for peers to stay in the session until most if not all peers get the file and which balances the resource consumption among peers. BitHoc does not sacrifice this point compared to standard BitTorrent due to the few pieces it sends far away in the network and by keeping the *tit-for-tat* algorithms of BitTorrent unchanged. More information about our research, our experiments and our GPL licensed open-source code can be found on this web site [5].

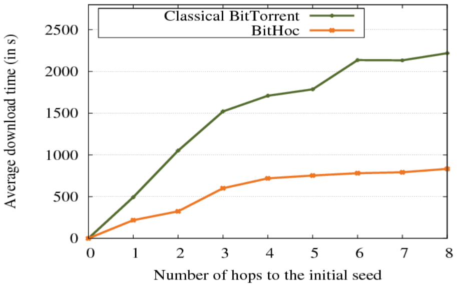


Fig. 4. Experimentation results

V. Conclusions

In this demo proposal, we describe our solution for content sharing in wireless ad hoc networks. The architecture that we propose contains two components: a distributed membership management service and an optimized content sharing service. The design of these services takes into consideration the constraints of mobile environments and the user needs. With the help of a real test-bed composed of PDAs and smartphones, we were able to validate our solution and to compare it to other classical approaches. The experiments show that our application outperforms the classical approach and highlight the utility of its features. In particular, we are able to reduce by a factor of 2 to 3 the download time.

References

[1] BitTorrent protocol. http://wiki.theory.org/BitTorrentSpecification.

[2] M.K.Sbai, C.Barakat, J.Choi, A.Al Hamra, T.Turletti, ”Adapting BitTorrent to wireless ad hoc networks” in proceedings of Ad-Hoc Now 2008, Sophia Antipolis, France.

[3] http://www.alhem.net/project/btpeer/

[4] http://www.grc.upv.es/software/intro_olsr.html

[5] http://planete.inria.fr/bithoc/