A Management Service for P2P Content Over MANets

Marcelo H. Vithoft   Altair O. Santin   Luiz A. de Paula Lima Jr.
Pontifical Catholic University of Paraná (PUCPR)
Graduate Program in Computer Science (PPGIa)
Curitiba – PR – Brazil
{vithoft, santin, laplima}@ppgia.pucpr.br

Abstract — **A Mobile Ad Hoc Network (MANet) is a flexible way to interconnect mobile devices. Peer-to-peer (P2P) networks are serverless self-organized infrastructures that enable end-to-end communications. The P2P churn and the instabilities of MANets interconnections bring important challenges to P2P over MANet (PoM). A consumer peer in PoM wastes several of its scarce resources managing the communication with the peer that provides content. Our proposal aims at reducing the consumer peer control over the communication so that to avoid unnecessary consumption of its resources. A Virtual Content Provider is used for this purpose and a prototype was implemented in order to show the feasibility of the proposal.**

Keywords — **Content Integrity in PoM, Communication Instabilities Mitigation in PoM, Content Management.**

1. Introduction

Peer-to-peer (P2P) networks are characterized by the highly availability of the provider peers, but also by the frequent network joining and leaving of such providers, known as churn. Therefore, P2P network composition is very dynamic, and the consumer peer should be prepared to deal with the frequent changing in the content provider [1].

Mobile Ad-hoc Networks (MANets) offer great flexibility in terms of network configuration, but on the other hand they are very unstable in maintaining the connectivity [2].

Mobile devices (handhelds, smartphones, etc.) are capable of wireless communication in MANets despite their limitations in terms of processing power, storage and bandwidth.

Mobile devices dealing with P2P over MANet environments have to be concerned both with the dynamicity of the P2P network and with the instability of the interconnection between mobile devices in MANets [2]. Thus, the provision of communication support to minimize the negative effects of ad hoc operation mode and the provider peer's unavailability in the end-to-end connections with consumer peers is challenging.

Traditionally, P2P try to overcome the negative effects of providers' unavailability, during the content download by allowing parallel download from different source providers. In such a case, if the file content is not exactly the same in all the different provides, the offset in a file will be distinct from the others. Thus, when the entire file rebuild is done, the resulting content file will be corrupted.

A large amount of contents available in traditional P2P networks is polluted, fake or corrupted (P2P garbage) [3]. Therefore, mobile devices that consume content from the P2P network may spend an important amount of their resources establishing and maintaining connections with peers that may be offering garbage. Besides, the peers may be absent during content providing, or the connectivity in the MANet can be lost.

In a traditional P2P end-to-end connection, when the connection is lost, the consumer peer (P2P client software) maintains the status of the download progress. When the connection is reestablished, the P2P client software tries to find a provider for the same content. If there is P2P garbage announced in the P2P network with the same file hash and keywords, it could be selected by the consumer to carry on the download and the consumer will only be able to realize that the file content is corrupted at the end of the download. The consumer does not have any mechanisms to avoid such a scenario in traditional P2P networks, given the fact that it easy for a malicious P2P to make a fake announcement and provide garbage pretending be an authentic content.

This work presents a proposal to minimize the impact of the instability and dynamicity of P2P over MANets offering good availability for providers/content, integrity to the contents, and mechanisms that allow automatic management of contents and of end-to-end communication, facilitating the consumer peer action in a PoM.

The remaining of this paper is organized as follows. Section 2 briefly discusses relevant aspects of overlay, MANet and P2P networks. Our proposal is then introduced in Section 3. In Section 4, the architecture proposed is considered in a particular scenario and, in Section 5, the prototype implemented is presented. Section 6 consists of the evaluation of the prototype. Finally, Section 7 gives an overview of related works and conclusions are drawn in Section 8.

2. The Overlay, Mobile Ad Hoc and Peer-to-Peer Networks

The overlay tier creates a logical abstraction of the physical network, allowing the transposition of physical domains of the underlying network [4].

This work has been partially supported by Brazilian National Research Council (CNPq), grant nº 550962/2007-7.

The end-to-end communication between two computational entities without the need of specific servers to provide some resource is called a peer-to-peer network (P2P). A peer can play the role of both resource provider and consumer. P2P networks are mainly characterized by the dynamicity of peers connection/disconnection, decentralization, self-organization and scalability. A P2P network is generally developed over an overlay network.

The JXTA project [5, 6] has developed a framework to interface P2P systems, which is independent of programming language and transport protocol. JXTA for mobile devices is based on the Java programming language, and specifically on the Java 2 Micro Edition (J2ME) framework. The porting of JXTA over J2ME is called JXME.

The DHT (Distributed Hash Table) is implemented as an overlay network and consists basically of a set of tuples relating keys and values in the pair (key, value) [7]. Each node of a DHT is responsible for the storage of a set of entries, obtained from the hash function applied to the values and resulting in the respective keys. Thus, the DHT can work as a distributed index repository. The DHT also maintains a routing table with information about the known neighboring nodes in the overlay network.

JXTA adopted a hybrid solution, combining the use of the DHT loosely-consistent approach with limited-range rendezvous walker [8]. This strategy reduces the reply delay about 50%, in searches between rendezvous (a peer with special functions, e.g. index for the announced content) of the JXTA network [9].

A MANet (Mobile Ad hoc Network) is characterized by the absence of fixed base radio station infrastructure. Therefore, the network is composed of nodes that sometimes routes messages from a neighbor to another and sometimes send or receive data through their neighbors [10].

3. Proposal

Our proposal involves the creation of a P2P Virtual Content Provider (VCP) that releases the mobile peer – usually playing the role of a consumer in the P2P network – from the task of managing end-to-end connections with the peer providing the content. This scheme aims at hiding the dynamicity (churn) of the P2P network from the consumer peer, so that the content is transparently provided. Thus, we consider that VCP provides, transparently, a service for management of content on the PoM (Peer-to-peer over MANet).

The VCP can be any mobile peer operating in ad hoc mode, but in general it is a border peer, linking a MANet to a wired or wireless infrastructure network. This case is more common since there are more contents available on traditional P2P networks than in MANets. In spite of the churn, P2P providers in wired / wireless infrastructure networks are more stable than those in MANet [11].

In practice, the consumer peer does not realize that it gets connected to the VCP. From the point of view of the consumer peer, it is requesting contents from an ordinary provider peer. In fact, the VCP can intermediate all the connections between a peer consumer and the peers that effectively provide the required content. All requests from a MANet pass through the VCP and, therefore it can act in two modes, namely, passive or active.

A VCP acts in passive mode when it receives a request for some contents and only forwards the message to the Content Index based on the DHT (CID). Such a behavior can be motivated by the fact that VCP searched its catalog and found an entry for such content, which is in cache and it has already been announced on the CID. Therefore, the request would find the VCP announcement for such content on the CID.

A VCP operating in active mode includes locating several content providers, downloading the content from some of such providers, classifying/filtering the content, and the announcing the VCP itself as a provider for the content. The location of provider peers involves obtaining at least P providers, candidates, to be the servers for the required content; the P value is configured by each VCP administrator.

After the VCP has completely downloaded the contents from P providers, the contents classification engine (CE) computes the hash function of each file content. It then compares each hash code obtained with the respective hash code announced in the CID. At the end of the process, the CE chooses a subset (CS) from the set P of providers. In other words, the subset CS contains only the providers which the file content and corresponding hash code match. If more than one kind of providers match in CS, the criterion used to select one set of providers with the same match is to choose the set with the most significant number of providers. However, if there is an equality of providers in different sets, the selection is at random. Therefore, the VCP cannot be held responsible for possibly inaccurate evaluation of the automatic content classification. Even so, it has been estimated that the adoption of VCP in PoM mitigates the amount of corrupt content files.

The CE in fact acts as a filter, aiming at preventing fake announcements of contents in the P2P network – an announced hash code that does not match the one computed for a given file content. Therefore, in the proposal, the content integrity assured by VCP is achieved through the filtering mechanism applied by the CE.

The hash code computed by the VCP that denotes the CS subset is stored in a local hash table (LHT). The data stored in each entry of the VCP local hash table are a list of keywords announced in the P2P network associated to the file content. The goal of these keywords storage is to eliminate equal content files announced in the P2P network with different keywords, but the same content. In other words, the VCP stores only a copy of a file content when a hash code resulting from CE already exists in the LHT. However, in case the keywords have not yet been include in the list (the LHT), the respective entry will have its keywords list updated to maintain its consistency.

The VCP local hash table works as a catalog for file contents available locally. Each time a content is provided by the VCP, after the CE choice from CS, the content is maintained in the VCP local storage for cache purposes.

All the contents available in the VCP cache are announced in the Content Index based on DHT. Thus, a VCP for a given content can be easily found through the CID.

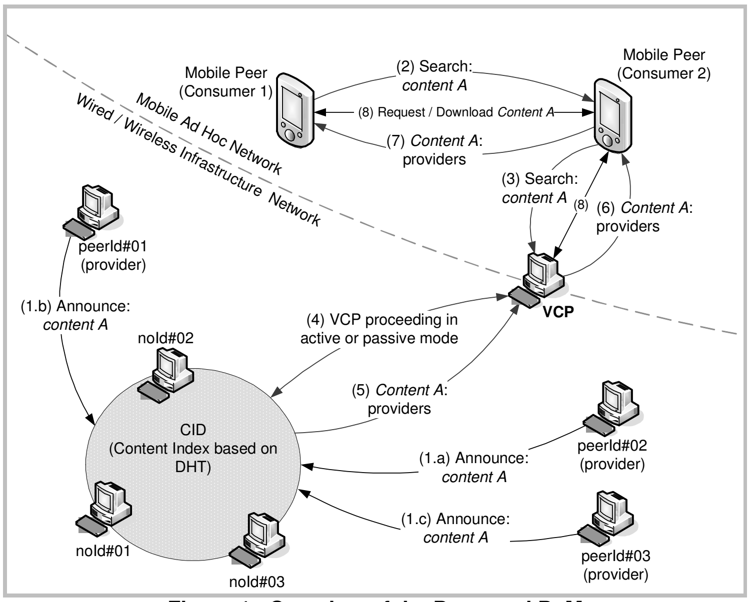

Figure 1. Overview of the Proposed PoM

When providing some contents, the VCP maintains the consumer's connection state. This makes possible to retrieve the exact file content offset that has already been download successfully by consumer peer. This can be useful after a MANet disconnection that happened suddenly, without the knowledge of the consumer peer.

Figure 1 shows the behavior of the proposed PoM environment. First of all, it is assumed that the providers have previously announced the available content in the DHT (events 1.a, 1.b and 1.c). When, the consumer peer requests a content from the CID, the VCP intercepts the communication that came from the MANet (event 2 and event 3), looks up for the required content locally and if the content is found, the search is forwarded to the CID directly (event 4).

On the other hand, when the required content is not available at the VCP, it starts its active mode of operation, depicted in Figure 2. Firstly, the VCP searches for the content on CID (event 4.a) receiving back a list of providers for the required content (event 4.b). The VCP requests content from *P* providers (event 4.c and 4.d). Notice that, in this example, provider#01 fails and the VCP starts the download from provider#03, to maintain the required number of *P* providers.

After the VCP completes the download of content (events 4.e and 4.f), the classification engine (CE) is invoked and the content resulting from the classifier is stored locally. Next, the VCP announces itself as a provider for the required content in the CID (event 4.g) and forwards the search request coming from MANet (event 4.i). Then, VCP receives back the provider candidates, now including itself as a provider of intact content (event 5, Figure 1). The last event is propagated through the MANet up to the consumer (event 7, Figure 1). Then, the consumer evaluates the provider candidates for the required content and normally chooses to download it from the VCP (event 8), given its good attributes as a provider.

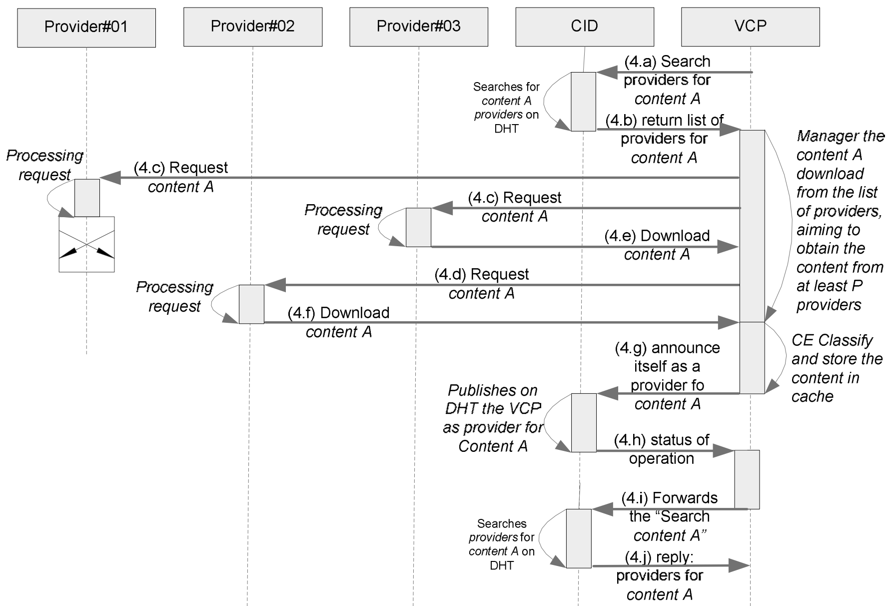

Figure 2. VCP in active mode

4. Scenario

Media files (like audio, video, and so on) are very common contents in traditional P2P networks. However, no guarantee is generally given about their integrity and availability. The goal of our proposal is to provide these guarantees in a transparent way for consumer peer.

When a consumer peer searches for a keyword in traditional P2P network it receives back a list of providers, depending of the P2P client (front-end). Then, the consumer can choose one provider from the list, based on the provider's bandwidth, average time of availability, etc. But, these kinds of attributes do not guarantee availability in MANets, neither do they guarantee integrity of file contents. Therefore, the goal of the VCP is to implement the necessary mechanisms to provide those guarantees. Although this is only one simple example, there are many other scenarios in which the proposed approach could be applied.

Figure 3 shows the behavior of the proposed entities when the VCP provides the required content from its own cache in passive mode. The scenario is based on a subset of the entities presented in Figure 1. The scenario assumes that the contents, once downloaded a first time, are classified, the keywords and hash code are stored in the catalog, and the content classifier engine selected the most probable intact file content retrieved from the list of providers and stored it locally in a cache file.

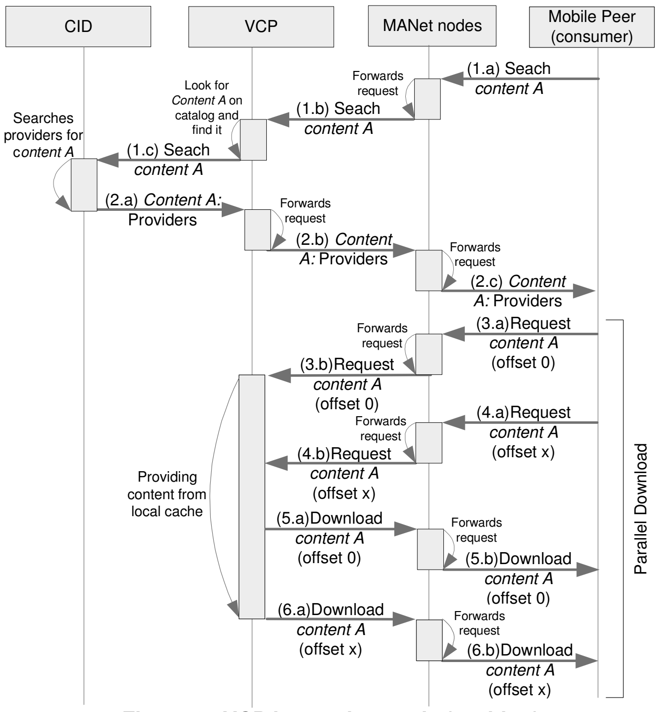

Figure 3. VCP in passive mode (caching)

In Figure 3, the consumer peer searches for a content from the CID. The request is forwarded through the MANet node to the VCP (event 1.a, 1.b and 1.c). Notice that, when the VCP receives event 1.b, it checks its catalog. If the requested content is found, the search is just forward to the CID. However, if the content is not found in the VCP catalog, another procedure is started in VCP as depicted in Figure 2. After receiving back the reply to the search from the CID (event 2.a), the VCP forwards it, believing that through its native features (attributes), it appears in the top of ranking list of provider candidates to serve the content in the consumer peer GUI (Graphical Unit Interface).

The consumer peer decides to make a parallel download, given the excellent features presented by the VCP as a provider. Thus, the consumer peer requests the offset 0 of the content file (event 3.a) and another offset (event 4.a). After the download was completed (event 5.b and 6.b), the file content can be rebuilt on the consumer peer through the correct concatenation of offset parts. Besides, the integrity of the entire file content can be checked against the hash code stored in the corresponding catalog entry.

Notice that the parallel download can be used in a more reliable way than the traditional one, since the file content integrity can be assured by the VCP as well as the status of connection. Thus, if the communication is suddenly interrupted, it can be restored from the same provider (the VCP). This strategy avoids file corruption common in the traditional approach, on which it is only possible to find out that some contents are corrupted after it has been completely downloaded.

5. Prototype

The prototype makes use of the JXTA infrastructure and the NetBeans IDE 5.5 [12] with the 2.5 toolkit for wireless communication [13]. This toolkit emulates mobile devices, providing JXME support to the prototype.

The proposed prototype architecture (Figure 4) is composed of a mixed network. The CID is on a traditional P2P JXTA overlay network located in a wired/wireless infrastructure network and the VCP can access directly both the CID and the MANet. The CID is super-peer of the JXTA network – a peer with special functions, namely rendezvous. One of them is to index the content announced in the JXTA network through a DHT-based loosely-consistent network of rendezvous [8].

The VCP plays the role of a P2P proxy in order to maintain the compatibility with mobile P2P consumers connecting from MANets, using JXME. This is how the VCP as a proxy performs caching of download contents.

The VCP administrator is responsible for setting up the time span within which the content will remain in cache and the amount of memory destined to the storage of contents in cache. In spite of the importance of such updating policies and caching maintenance techniques, they are out of the scope of this work.

In the prototype, the integrity is obtained through a hash function applied over a content. When the goal is to verify the integrity, the resulting hash code is compared with a previously announced hash code; if they match, the integrity of both is verified. When the goal is to provide integrity to a content, after computed, the hash code is announced on the CID (Content Index based on DHT) together with the keywords.

In Figure 4, all blocks identified as APP correspond to the implementation to support the proposed architecture. The VCP APP provides an interface to access the proposed architecture facilities. One part of the facilities is addressed to administrator of the VCP, *VCPAdminFacilities*, and the other one to consumer peers, *VCPConsumerFacilities*.

The VCPAdminFacilities is used to setup JXTA and JXME file/parameters, and to handle the catalog and the cache. The VCP configurations are taken into account only at startup.

The VCPConsumerFacilities is the "front-end" of VCP for the consumer peers. In fact, consumer peers are unaware of their interaction with the VCP (event 1). Since the VCP intercepts the requests for contents, makes it available locally, and appears as a provider with good attributes in the P2P client GUI. Therefore, the consumer peer is induced to choose the VCP appearing on top of the list of provider candidates to serve the required content.

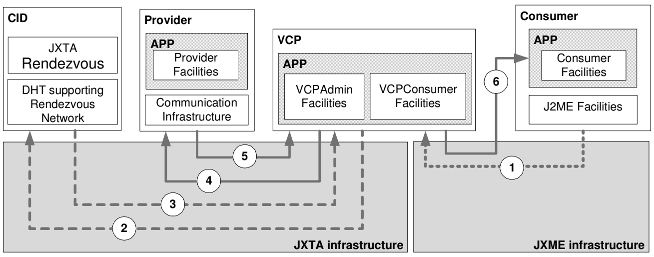

Figure 4. Prototype Overview

The VCPConsumerFacilities implements the searcher and downloader, classifier, cataloger, and connection manager modules. The searcher is responsible for locating providers for the content (event 2). The search request is routed through the DHT-based rendezvous network inquiring the VCP catalog module through the JXTA. The downloader module controls the searcher in order to maintain the number of *P* providers, which is required by the classification of contents. It also downloads the contents from the *P* providers found by the searcher (event 4 and 5). The classifier implements the contents classification engine (CE) and estimates which content is the more likely to be authentic; the classifier feeds the cache of contents. The cataloger module records all contents outputted by the classifier. Finally, when a connection is established with the VCP, it logs information like the peer id, the number of parallel download sessions, the offset and the amount of already download content in each session to recover a connection if it breaks.

Although provider and consumer peers are detailed separately for clarity, in practice, each peer plays both roles according to their current needs.

6. Prototype Evaluation

From the qualitative point of view, the proposed VCP tries to guarantee that the contents downloaded by a mobile application are intact, i.e., the application will not waste time downloading corrupt or unexpected contents. Besides, if an interconnection instability happens, the download session recovery is facilitated by the VCP control over the consumer peer status.

The VCP decreases significantly the file content corruption in parallel downloads, so the integrity of the content is taken into account as aforementioned. Also, the VCP is a more stable provider and the distinct offsets of the parallel download are related to the same content file. This approach is more reliable than multiple downloads from different providers adopted in traditional clients of P2P, mainly due to the fact that there is no guarantee that the file contents are the same, although the announcement in the CID appears as the same.

Aiming to measure the impact of the insertion of the VCP into the proposed PoM, from the quantitative point of view, we considered a simple but intuitive scenario (Figure 5). In such experiment, let *tR*, *tR'*, *tB'*, *tV*, and *tB* be time frames, considered in average after 100 measures, concerning the search and content download operations. Specifically:

- tR = amount of time spent in a consumer search request to the VCP;
- tR' = amount of time spent in a VCP search request to the CID;
- tV = amount of time spent by the integrity check, catalog record, announcement in CID (VCP internal procedures);
- tB = amount of time spent downloading a file content from VCP to the consumer peer;
- tB' = amount of time spent downloading a file content from provider peer to VCP;

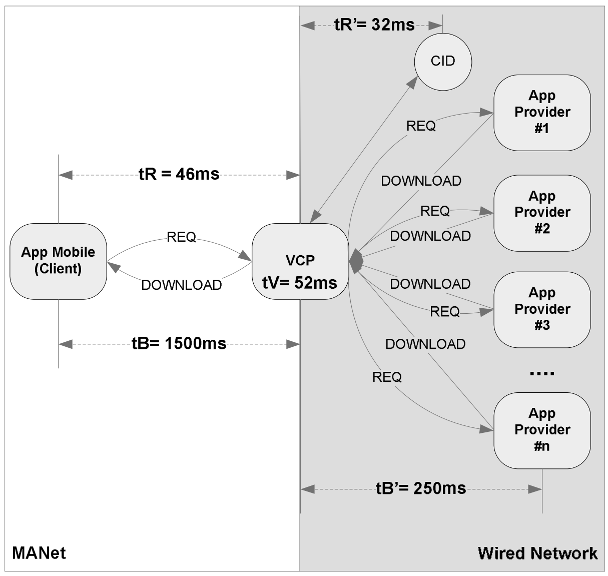

Figure 5. Evaluation Scenario

Supposing that the consumer accesses directly the provider peer (Figure 5), the total time spent to download a file content takes *td*:

td = tR + tR' + tB' + tB.

Therefore, based on the time frames obtained through the essays with the prototype, in order to download some contents directly from a provider, *td* is:

td = 46ms + 32ms + 250ms + 1,500ms = 1,828 ms.

Now consider the usage of the VCP, as depicted in Figure 5. When the VCP intercepts the request and obtains the list of providers, it maintains *P* downloads in parallel, which does not increase the total download time from the providers. Therefore, *tv* is given by:

tv = tR + tR' + tB' + tV + tB.

Thus, in spite of the parallel download from *P* providers, the classification of content, and the other aforementioned activities performed by VCP, only the time for processing in VCP is added to the total amount of time for the consumer. That is:

tv = 46ms + 32ms + 250ms + 52ms +1,500ms = 1,880 ms.

In other words, from the measures obtained from the prototype, the inclusion of the VCP represents around 3% of the total amount of time spent in the direct downloading from the provider to the consumer. Nevertheless, if all advantages of the use of VCP are taken into account and if it is possible to download content in parallel (in case there is enough bandwidth in the consumer endpoint), then tv can be reduced and the usage of the VCP is significantly advantageous if compared to traditional P2P.

7. Related Work

Next, we briefly discuss some of the related technical works found in the literature.

In [11], the simulation detected that the ability of the Chord DHT to perform consistent queries in MANet was badly affected by such environment. The authors argued that that happened due to the pessimistic strategy adopted by the simulator algorithm, which simply terminates the pending queries by time-out. The results were nevertheless inconclusive.

In [14], an implementation framework is proposed whose goal is to hide the connection instability from P2P application developers. The implementation requires changes to the core of JXTA in order to include functionalities like management of intermittent connections and multiple physical interfaces. These changes increased the mechanism for resources discovery.

Our proposal aimed to mitigate the connectivity problem, but without change the JXTA, because we intended that our proposal would maintain compatibility with currently used P2P applications. The DHT is adopted by JXTA network but we applied it only in the wired network, since, according [11], DHTs in MANets could produce undesired results.

8. Conclusions

This work presented a proposal to minimize the impact of the instability and churn of P2P over MANets in a scenario able to offer good availability for providers/content, integrity to the contents, and a service to automatic management of content and the end-to-end communication, facilitating the consumer peer action in PoM.

The proposed scenario takes into account the automatic classification (filtering) of contents, obtained from traditional P2P network and stored at the VCP cache, enabling availability and integrity.

The VCP playing the role of a P2P proxy maintains the compatibility with current P2P content consumer and makes a bridge between MANet and wired or wireless infrastructure networks.

The VCP is responsible for locating peers that provide a given content, for managing the end-to-end connection between consumer peer and itself and between the provider peer and itself, for maintaining the connection/session state with the consumer peer, and for providing an intact content to consumer.

The prototype developed showed the viability and easy integration of the proposed architecture with the traditional P2P networks over MANets.

REFERENCES

[1] Schollmeier, R., "A definition of peer-to-peer networking towards a delimitation against classical client-server concepts", In Proceedings of the 7th EUNICE Open European Summer School, Paris (France), September, 2001, pages 131-138.

[2] Perkins, C. E., "Ad Hoc Networking", Addison-Wesley, Boston, MA, USA, 2001.

[3] Kumar, R., Yao, D., Bagchi, A., Ross, K., Rubenstein, D., "Fluid Modeling of Pollution Proliferation in P2P Networks", In Proceedings of ACM Sigmetrics, 2006, pages 335–346.

[4] Andersen, D., Balakrishman, H., Kaashoek, F. & Morris, R., "Resilient Overlay Networks", In Proceedings of 18th ACM Symposium on Operating Systems Principles, Banff, Canada, October, 2001.

[5] Traversat, B., Abdelaziz, M., Doolin, D., Duigou, M., Hugly, J.-C. , Pouyoul, E., Sun Microsyst., Inc., "Project JXTA-C: Enabling a Web of Things", In Proceedings of the HICSS-36 Conference, January, 2003.

[6] Oaks, S., Traversat, B., Gong, L., "JXTA in a Nutshell", O'Reilly Press, 0-596-00236-X, September, 2002.

[7] Stoica, I., Morris, R., Karger, D. R., Kaashock, M. Frans et Balakrishman, H., "Chord: A scalable peer-to-peer lookup protocol for internet applications", In Proceedings of the ACM SIGCOMM, San Diego, California, August, 2001, pages 149-160.

[8] Traversa, Bernard, Abdelaziz, Mohamed, Pouyoul, Eric, "Project JXTA: A Loosely-Consistent DHT Rendezvous Walker", Sun Microsystems, Inc, San Antonio Road, Palo Alto, CA, 2003.

[9] E. Halepovic, R. Deters, and B. Traversat, "Performance Evaluation of JXTA Rendezvous", In Proceedings of International Symposium on Distributed Objects and Applications (DOA '04). Agia Napa, Cyprus: Springer Verlag, October 2004, pages 1125–1142.

[10] van der Merwe, J., Dawoud, D., and McDonald, S., "survey on peer-to-peer key management for mobile ad hoc networks", In ACM Comput. Surv. 39, 1, Article 1, April, 2007.

[11] Cramer, C., Fuhrmann, T., "Performance Evaluation of Chord in Mobile Ad Hoc Networks", In Proceedings of MobiShare'06 - ACM; Los Angeles, California, USA, September, 2006.

[12] "NetBeans IDE Home", http://www.netbeans.org/, Last Access: January, 2009.

[13] SUN Developer Network (SDN), "Sun Java Wireless Toolkit for CLDC Home", http://java.sun.com/products/sjwtoolkit/, Last Access: January, 2009.

[14] Bisignano, M., Calvagna, A., Di Modica, G., Tomarchio, O., "Expeerience: a Jxta middleware for mobile ad-hoc networks", In Proceedings of 3rd International Conference on Peer-to-Peer Computing (P2P'03), IEEE, 2003.