HyCast—Podcast Discovery in Mobile Networks

Adrian Andronache
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
(+352) 46 66 44 5205
adrian.andronache@uni.lu

Matthias R. Brust
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
(+352) 46 66 44 5205
matthias.brust@uni.lu

Steffen Rothkugel
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
(+352) 46 66 44 5205
steffen.rothkugel@uni.lu

ABSTRACT
Podcasts are a popular way to provide multimedia information about certain topics. A multitude of podcast servers exist in the Internet, allowing people to subscribe to them. Typically, podcasts are downloaded onto desktop computers and copied on mobile devices to be played while being on the move. In this paper, we extend the idea of podcasts, making them available in mobile network environments. In particular, HyCast does not rely on central podcast directories. Instead, HyCast also allows discovering, subscribing to, and downloading podcasts and episodes in the local neighborhood. For the dissemination of podcast information, we introduce and evaluate two different strategies. One is based on peer-to-peer communication between one-hop neighbors. The second one employs clustering to reduce the overhead of the podcast information dissemination.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Distributed networks, network communications, network topology, wireless communication; C.2.2 [**Network Protocols**]: Applications, Protocol architecture.

General Terms
Algorithms, Management, Performance, Design, Experimentation.

Keywords
Multimedia, podcast, mobile, clustering, peer-to-peer, content distribution.

1. INTRODUCTION
A podcast is a multimedia file distributed over the Internet using the syndication feeds mechanism. It can then be played on computers and mobile devices. The multimedia files provided by podcasts are comparable to recorded radio or television series. The provider—called podcaster—creates the content to offer and posts the information on a Web server. The location of the post is expected to be permanent. The information is posted as syndication feed in RSS format, which provides information about the series and its episodes such as publishing date, title, description and so on. It may contain entries for all episodes in the series, but is typically limited to a short list of the most recent ones.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WMuNeP'07, October 22, 2007, Chania, Crete Island, Greece.
Copyright 2007 ACM 978-1-59593-804-6/07/0010...$5.00.

The users interested in podcasts are called subscribers. A subscriber uses a software client called aggregator to subscribe and manage the feeds. An aggregator usually launches when the computer is started and runs in the background. It manages the set of feed URIs subscribed to by the user and downloads new episodes at a specified interval.

Podcasts were initially meant to allow individuals to distribute their own "radio shows" but meanwhile the mechanism is used in a variety of other scenarios like distribution of school lessons, audio tours of museums, conference meeting alerts and updates and alike.

The current technique allows the subscribers to search and to subscribe podcasts as well as to download episodes using an Internet connection. Usually the files are downloaded and played on computers or copied on mobile devices and played on the way. The current podcasting mechanism does not allow mobile device users to exchange episodes of subscribed podcasts or to recommend related and well-rated podcasts to other users in a mobile ad-hoc environment.

Podcasts application scenarios are numerous. For example, today podcasting is already applied to share class lectures to students at schools or Universities [1, 9] and conference presentations to conference attendees that visit a parallel session [11]. Even for manual support and consulting podcast can be successfully applied [10].

In this work we introduce the HyCast application which uses the podcasting mechanism but also augments it by enabling an "on the way" supply of yet unknown podcasts or new episodes of subscribed podcasts among connected devices in mobile ad-hoc networks.

This paper presents, evaluates and compares two mechanisms used by HyCast to disseminate the information about podcasts available in the mobile network. This information is required by devices with shared interests in order to build groups and exchange podcasts episodes. The information is also used by the application to discover yet unknown podcasts related to the taste of the user. Exploiting the network topology in order to streamline information exchange and minimizing overhead is one of the main goals.

The remainder of this paper is organized as follows. In Section 2 related work is described. Section 3 introduces the HyCast application scenario. Details about the HyCast network management and information exchange mechanisms are presented in Section 4. The simulations and the results are described in Section 5. The conclusions and the future work in Section 6 finish this paper.

2. RELATED WORK

In [7] Görgen et al. present a communication pattern for information dissemination in highly mobile ad hoc networks. The authors have implemented, tested and evaluated this pattern in an example application called UbiQuiz. This application is a mobile quiz game in the style of "Who wants to be a millionaire". Whereas the devices running that application are able to efficiently exchange questions. The general idea of the underlying communication pattern is to use the short interaction period of passing devices to efficiently synchronize the information of each device. This is realized by creating single hop peer-to-peer overlays of interest domains, where devices with the same interest domain are detected via beaconing. To maintain the network load at minimum only devices interested in same domain try to synchronize their information. Furthermore the amount of data transferred is reduced by exchanging profile information first. The authors term this mobility driven information synchronization as en-passant communication.

Bo et al. [3] describe two probability-based schemes for multimedia content location in wireless ad hoc networks. The idea of the presented approaches is to predict the location of multimedia data sources in the network based on query history and probability rules. The authors aim at designing a content-based retrieval scheme that employ data content distribution knowledge in order to optimize the search cost, response time, and system overhead. The proposed schemes are motivated by the observations that (a) a node may have certain interests and query some multimedia data items frequently; (b) the data contents of a node are often semantically similar to its queries; (c) some nodes in the ad hoc network may have shared interests and generate similar queries. The authors propose to exploit the query history of mobile nodes to find the relationships among multimedia data objects, and use this knowledge to determine the data contents of mobile nodes.

Lau et al. [8] present a cooperative caching architecture suitable for continuous media (CM) proxy caching in mobile ad-hoc networks. The proposed scheme introduces an application manager component, which is interposed between traditional Internet CM applications and the network layer. The application manager transparently performs data location and service migration of active CM streaming sessions so as to exploit nearby data sources based on the dynamic topology of the ad-hoc network. The presented schemes actively check the network periodically to locate proximal information sources. Whenever closer information sources are found, the service migration facility is invoked to switch to the new source.

In our previous work the HyMN System [2] was designed to provide multimedia files to mobile devices using hybrid wireless networks. The HyMN clients on the mobile devices establish clusters using the WACA algorithm [4] by electing local leaders called clusterheads. The clusterheads register at the HyMN backbone via a cellular network and retrieve the available information domains—called Interests—which are broadcasted to the cluster members. An Interest can be for instance a sports event from which the HyMN backbone will provide multimedia highlights. Each cluster member can select its Interests that will be sent to the clusterhead, which registers them at the backbone. When the backbone can provide fresh multimedia files it will push them to the clusterheads, which will multicast the received files to the interested slaves.

A drawback of the HyMN system is that the clients are able to receive information only from the central HyMN backbone, since it is the only provider. The backbone needs to keep track of all clusters and their interests and to manage and provide multimedia related to several domains, which requires a high amount of storage and communication resources. In order to optimize the resources usage, the central backbone will provide only information related to mainstream interests. Thus, the system is not very well suited for small marginal interest groups or those of local importance only.

The backbone is providing new multimedia files related to an Interest using a push approach, called injection [5]. As mentioned before, when a new item is available the backbone will inject it via cellular links to all clusterheads that have registered the appropriate Interest. The disadvantage of this mechanism is that the clusterheads being out of cellular coverage for a period of time and those that register later will miss injections, i.e. new multimedia items.

3. THE HYCAST APPLICATION

3.1 Multimedia providing by Podcasts

In order to overcome some of the HyMN drawbacks, HyCast (Hybrid Network Podcasting Application) was developed.

The HyCast application is prototypically implemented on top of the JANE simulation environment [6]. JANE is designed to support application and protocol design in the realm of ad-hoc networks. One distinct feature of JANE is that applications can be executed on top of simulated devices as well as in so-called platform mode on real hardware.

HyCast is intended to run on mobile devices with Wi-Fi or Bluetooth adapters like e.g. notebooks, PDAs or mobile phones. Currently many of these devices are also equipped with a cellular adapter that enables HyCast to access resources from the Internet or interconnect devices in different network partitions.

A main distinction between the two applications is that HyCast is using the popular, easy to deploy podcasting mechanism to provide the multimedia items to the mobile devices. The HyMN Interests are modeled as podcasts and the newly available multimedia will be provided as podcast episodes. Thus, the devices are able to check for new items and pull them either from connected ad-hoc devices in the vicinity or from the backbone if an Internet connection is available.

By using the podcast mechanism everyone can be a podcaster and the clients are not forced to retrieve the information from a single central provider. This makes the application adequate also for small marginal Interest groups. The podcasts can be provided even from Web servers running on mobile devices. For instance, a teacher can offer a podcast containing learning material for the current lecture from his notebook. Thus, the aggregator of the subscribing students will detect and pull the new lecture episodes during the class making them available instantly, even in case no Internet connection is available.

In order to avoid situations where devices are in charge of providing items from the backbone, which are not matching the own Interests, HyCast builds groups of devices interested in the same podcasts. The so-called podcast groups will be organized in clusters by using the information provided by the WACA algorithm. The device with the highest weight will act as clusterhead within a local podcast group. Devices that feature a backbone connection—via cellular network or access point—have an increased weight, thus being preferred to become clusterheads. A clusterhead is in charge of maintaining an uplink to the podcast Web server and keep track of new episodes. The members of a podcast group can also exchange new or missing episodes within the ad-hoc network, thus unloading the cellular base stations, access points and the podcast Web servers.

If the user subscribes a podcast on the way, the application joins that particular podcast group in the ad-hoc network and tries to retrieve the episodes from the neighboring group. If this operation is not successful then the episodes will be downloaded from the podcast server as soon as an Internet connection is available.

3.2 The “Big City Life”-Scenario

The HyCast application will be illustrated by a scenario we call Big City Life. Assume residents and tourists of big cities where many people—potential mobile device users—meet at train or metro stations, at universities and schools, in shopping centers, restaurants, and pubs. The shopping centers of the city provide podcasts about offers of the day that contain multimedia files showing the products. The restaurants are podcasting the menu of the day also containing multimedia about the food, drinks and location. The podcasts of the theatres are containing information about tonight’s shows and highlights and the podcasts of the discotheques are informing about events and party mottos presenting the DJ’s and the music that will be played.

The mobile device users in the city can subscribe to the podcasts of the favorite restaurants, theatres, shopping centers etc., thus being up to date about current offers and events. They can subscribe and download the podcasts on computers with broadband Internet connection and download them onto their mobile device, which is the current common procedure.

As already mentioned, the HyCast client aims to augment the podcast providing mechanism by enabling the mobile devices to exchange episodes of subscribed podcasts, to recommend similar, well-rated podcasts and to subscribe to new ones in an ad-hoc mobile environment. Additionally, devices with a cellular network interface are able to pull new podcast episodes from the Internet servers and share them locally within ad-hoc networks.

Imagine a traveling HyCast user meeting other people in metros, trains, shopping centers, pubs or restaurants. Some of them might have recently downloaded new podcast episodes either via cellular uplinks or by copying them from desktop computers onto their mobile device. The HyCast client on the traveler’s device will group up with the subscribers of similar podcasts and exchange new or missing episodes. Thus, the traveler will be able to receive podcasts on the way without having to use an Internet connection.

The HyCast client is also designed to use a collaborative filtering mechanism to find new unknown well-rated podcasts related to the ones subscribed already by the user. By meeting each other in the ad-hoc environment the HyCast users will be able to find out about other podcasts of interest and subscribe to them immediately, which saves them from an Internet search.

The HyCast client can be used for instance by a tourist which arrives in a big city and does not have knowledge about the local podcast providers. He will meet other tourists and residents with mobile devices at the airport, train or metro station, shopping centers, pubs and restaurants. The HyCast client on the tourist’s mobile device will search within the ad-hoc network for podcasts and provide the information to its user. Thus, the tourist is able to choose between the local podcasts about restaurants that offer the favorite food, locations and events he may like or shopping offers that can be interesting for him. By subscribing to the podcasts of interest found in the mobile ad-hoc network, the tourist will receive up to date information about offers and events in the city during the stay.

As mentioned already, when a user subscribes to podcasts while being part of an ad-hoc network, then the client will request the members of the certain podcast group for available episodes. Thus, the user can be provided with the requested episodes on the way, right after the subscription. Such situations can occur for instance at the airport, in a restaurant or in the hotel lobby. Sometimes the ad-hoc network does not provide paths to podcast group members that can provide missing episodes. In this case the HyCast client downloads them from the podcast server in case an Internet connection is available, like e.g. in the hotel room.

4. PODCAST DISCOVERY MECHANISMS

This section describes the mechanisms used by the HyCast application to exchange information about available podcasts between the devices in a mobile ad-hoc environment.

In order to reduce the control traffic, the mobile devices exchange only a list containing hash codes of the known podcasts URIs. This list is considerably smaller than a list containing the complete podcast feeds, which unloads the ad-hoc network. Each hash code is unique and will represent the ID of the podcast. When HyCast discovers unknown podcast IDs in a received list, the sender is requested for the podcast feeds. The IDs of the podcasts that are subscribed by the own user are marked by a flag.

The next sub-sections present two different communication patterns used by the HyCast application to disseminate the podcast information.

4.1 P2P Information Exchange Mode

The basic method for a device to gather information about available podcasts in the ad-hoc network is to exchange the podcasts list with each device in communication range. We will call this communication pattern P2P (Peer-to-Peer) information exchange.

When running in P2P mode, HyCast sends the own list of known podcasts to each device that enters the communication range. After the neighbor acknowledges that it has received the list, his device ID will be stored in the list of updated devices. This will avoid that the same podcast list will be sent more than once to the same device.

The exchange of the podcast lists together with the information about which podcasts are subscribed enables the mobile devices to discover shared interests and build local podcast groups in order to exchange missing multimedia episodes.

If a received list contains unknown podcasts, HyCast will update the own podcast list and clear the list of updated devices. This will lead to an exchange with the other devices in the network neighborhood and thus to a dissemination of the updated information.

The pseudo code of the P2P exchange algorithm is like follows:

```
N: Set of neighbor devices
U: Set of updated devices
L: List of known podcasts
Lx: Podcast list received from device x ∈ N

When received a beacon from d:
if d ∉ U then send L to d;

When received Ld:
if L\Ld ≠ null and d ∉ U then send L to d;
flag = false;
foreach p ∈ Ld do
    if p ∉ L then
            add p to L;
            flag = true;
    end if;
end foreach;
if flag = true then
    empty U;
    add d to U;
end if;
```

The algorithm is simple, easy to deploy and it disseminates new podcasts between mobile devices in a fast way. The main drawback of the algorithm is the sending of superfluous messages in some network settings. For instance the situation can occur where a mobile device d enters the one hop communication range of some devices which already have exchanged their podcast lists (Fig. 1a).

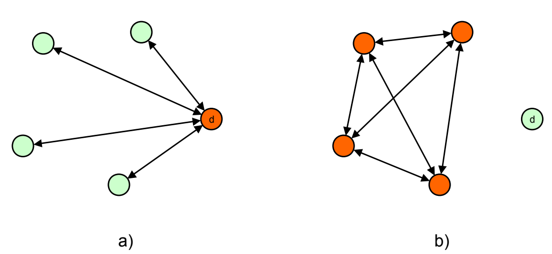

Figure 1. The new device will induce a superfluous exchange of podcasts list between the one hop neighbor devices.

The device d will exchange its own list with all devices, which will update their podcast list and clear their lists of updated devices. This will induce that each device will send to all other devices the own updated podcast list even if the lists are the same on all devices (Fig. 1b).

4.2 Cluster Information Exchange Mode

The second podcast information exchange algorithm employed by HyCast uses the cluster topology provided by WACA in order to reduce the number of unicasts, thus reducing the load and energy consumption of the mobile ad-hoc network [12].

4.2.1 The WACA algorithm

The WACA algorithm establishes an ad-hoc network topology that fits the needs of the applications running on participating devices. One objective of WACA is to avoid network communication overhead during the clusterhead election and clustering process. Therefore, the election of a clusterhead is based solely on information available locally. In order to achieve this, each device calculates—via a heuristic weight function—its own weight based on its device parameters like remaining power, memory load and number of neighbors as described in [4].

The weight is recalculated when changes of attributes occur. Each device propagates its own weight as part of the beacon, which is a periodically broadcasted message used in ad-hoc networks to detect devices in communication range. The algorithm only considers so-called neighbor devices, i.e. devices that mutually see each other. The devices run the algorithm each time the set of neighbor devices changes, e.g. when devices enter or leave the communication range, or when weights are updated.

Based upon the information about the neighborhood, each device elects the neighbor device with the highest weight as its clusterhead. This clusterhead also investigates its one-hop neighborhood, in turn electing the device with the highest weight as its clusterhead. This process terminates in case of a device electing itself as its own clusterhead due to the fact of having the highest weight among all its neighbors. We call all intermediary devices along such clusterhead chains sub-heads.

Each device propagates the ID of the own clusterhead by the beacon, which enables a device to detect whether it is used as clusterhead by neighboring devices.

More details about WACA and experiments results can be found in [4].

4.2.2 Information exchange

In the clustered HyCast network the clusterheads act as podcast information tracker. As mentioned above, the weight of the devices is based on remaining power, memory load and number of neighbors. Having the highest weight in the one-hop neighborhood makes a device the most appropriate candidate to act as a local information tracker.

When running in cluster mode, the devices are sending the podcast list only to the clusterhead. The device keeps track of the clusterheads that have received the current podcast list. This avoids that a list will be sent twice to the same clusterhead. When the podcast list changes then the list of updated clusterheads is discarded and the updated list is sent to the current clusterhead.

In mobile environments situations may occur when a device frequently changes the clusterhead during a short period of time, for instance when it crosses multiple clusters. In such cases the podcast list exchange with the clusterhead will not succeed and it should be avoided. In order to omit superfluous exchange attempts the devices assign a weight to the clusterhead link. When a new clusterhead is elected, the weight of the link to the old clusterhead is reset. The weight of the link to the new clusterhead starts to grow over time till it reaches a given *stability threshold*. The devices will send their podcast list to the clusterhead only when the weight of the link has reached the threshold.

The clusterhead stores the device that sends its list as a stable neighbor device and compares the received list with the own list. If the sender misses some podcasts, the clusterhead sends him the own list so that the new device can update its list. If there are unknown podcasts in the received list, the clusterhead updates the own list and send it to all known stable neighbors. Thus, the clusterhead will attempt to send the updated list only to devices that have a stable link to it.

The pseudo code of the cluster exchange algorithm is like follows:

```
N: Set of neighbor devices
S: Set of stable neighbor devices
U: Set of updated devices
c: The current clusterhead
t: The stability threshold
L: List of known podcasts
Lx: Podcast list received from device x ∈ N

When new clusterhead c is elected:
01: link_weight = 0;
02: wait 1 second;
03: link_weight++;
04: if link_weight < t then goto 02;
05: send L to c;

When received Ld:
01: if own device = c then add d to S;
02: if L\Ld ≠ null and d ∉ U then send L to d;
03: flag = false;
04: foreach p ∈ Ld do
05:     if p ∉ L then
06:             add p to L;
07:             flag = true;
08:     end if;
09: end foreach;
10: if flag = true then
12:     empty U;
13:     add d to U;
14: end if;
15: if own device = c then
16:     foreach d ∈ S do
17:             send L to d;
18:             add d to U;
19:     end foreach;
20: end if;
```

By receiving the information from the stable neighbors, the clusterheads keep track of available podcasts and also of podcast groups in the cluster. Each device in the cluster is now able to request the clusterhead for the list of devices subscribing a certain podcast, thus receiving the local podcast group. The list also contains the current weight of the devices and the information which device features an Internet connection—by using a cellular connection or an access point.

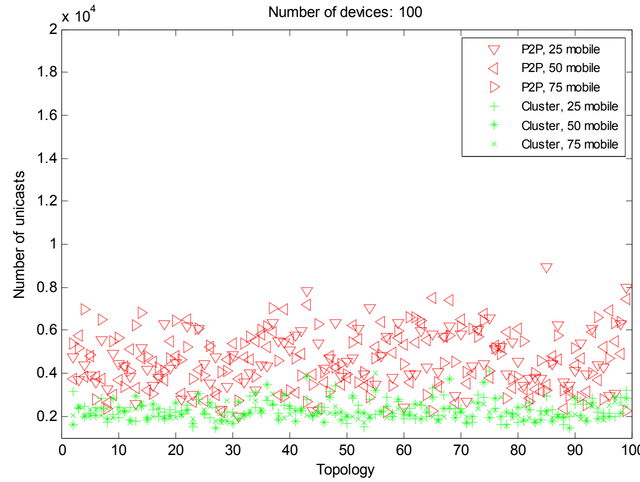

Figure 2. The overall number of unicasts is lower in cluster mode than in P2P mode.

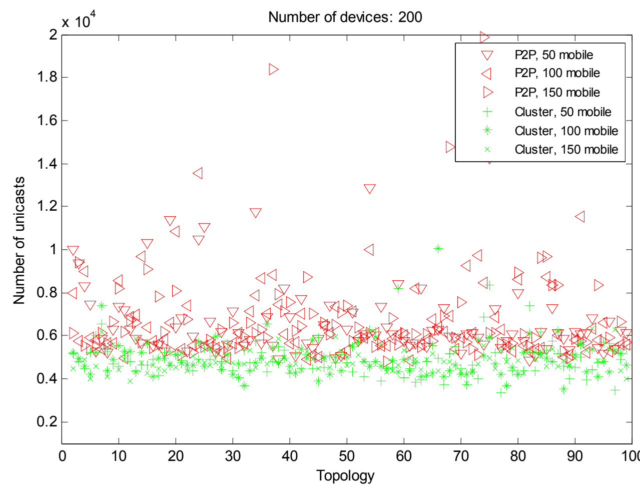

Figure 3. The overall number of unicasts is lower in cluster mode than in P2P mode.

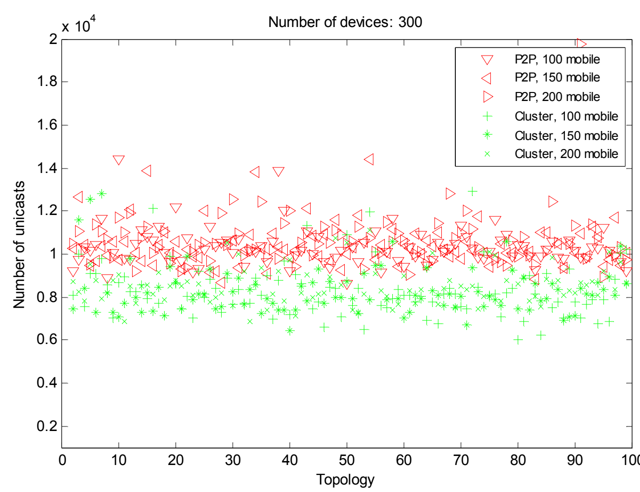

Figure 4. The overall number of unicasts is lower in cluster mode than in P2P mode.

5. SIMULATIONS

5.1 Simulation Settings

In order to determine the characteristics of those two information dissemination patterns, we performed simulation studies using the JANE simulator. Like already mentioned, one of the main goals is to streamline information exchange and minimize overhead and energy consumption by reducing the number of unicasts sent in the mobile network.

For the simulation we used an area of 300 m x 300 m and the transmission range of the devices was set to 40 m. A number of *N* nodes were deployed uniformly by random using a validated random generator initialized by independent seeds. To study the algorithms in sparse and dense networks, we conducted simulations with *N* set to the values 100, 200 and 300.

The HyCast application is designed for users at airports, shopping malls, university campus et cetera. In such environments some of the users walk around while others are staying at one place. To trace the impact of different rates of moving devices, we set a number *M* of device to be mobile while the other *N-M* devices are stationary.

We used the following values for of *M*:

- For *N* = 100, M = {25, 50, 75}
- For *N* = 200, M = {50, 100, 150}
- For *N* = 300, M = {100, 150, 200}

To model the mobility patterns of the mobile nodes we used Random Waypoint. The speed of the devices was set to 1.5 m per second (5.4 km/h), which is a common speed for human beings.

The clusterhead link stability threshold was set to 5 seconds.

Each of the abovementioned settings were used in simulation runs with 100 different random generated network topologies once in P2P and once in Cluster Mode.

For the experiments, each device started with one podcast unknown to the other devices in the network. To compare the two dissemination strategies, we traced how many unicasts need to be sent so that every device receives all podcast feeds. We also traced the time in order to analyze if there is a dissemination speed difference between the two communication patterns.

5.2 Simulation Results

In the first test set, we investigated the number of unicasts required for the *entire* podcast information to finally reach *all* devices.

The results are depicted in Figure 2, Figure 3, and Figure 4, for 100, 200, and 300 devices, respectively. In all three settings, and across all of the 100 different topologies used, cluster mode clearly outperforms P2P mode.

To illustrate the savings in terms of unicasts, the quotient of unicasts in cluster mode divided by unicasts in P2P mode is shown in Figures 5, 6 and 7. Taking the mean values, with 100 devices, cluster mode requires approximately half the number of unicasts compared to P2P mode. The mean values thereby are almost independent of the number of mobile devices. With 200 devices, a difference of about 22% remains. With 300 devices, the difference only drops slightly to about 19%.

However, there is a considerable impact of the mode chosen on the time required for the *entire* podcast information to reach *all* devices.

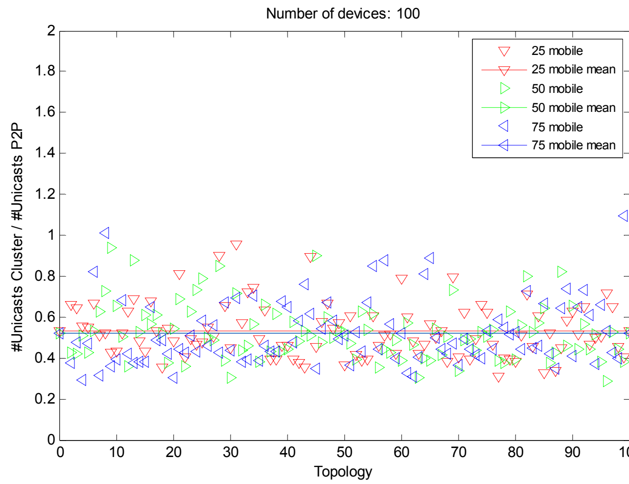

Figure 5. 100 devices: cluster mode requires approximately half the number of unicasts compared to P2P mode

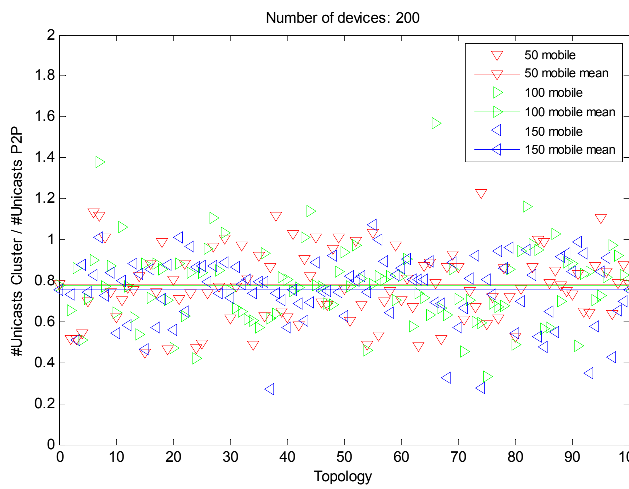

Figure 6. 200 devices: the cluster mode produces average about 22% lesser unicasts than P2P mode.

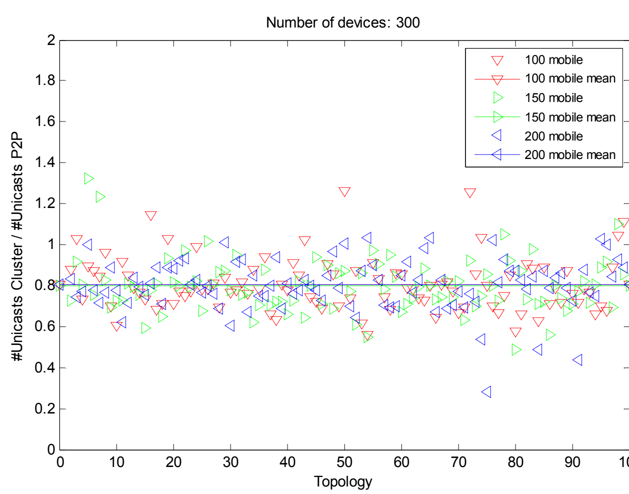

Figure 7. 300 devices: the cluster mode produces average about 19% lesser unicasts than P2P mode.

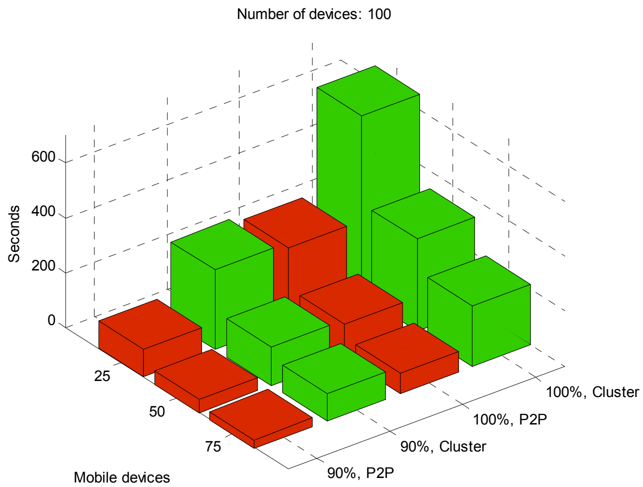

Figure 8. The information is spread faster when there are more mobile devices. The cluster mode spends about half of the time to disseminate the last 10% of the information.

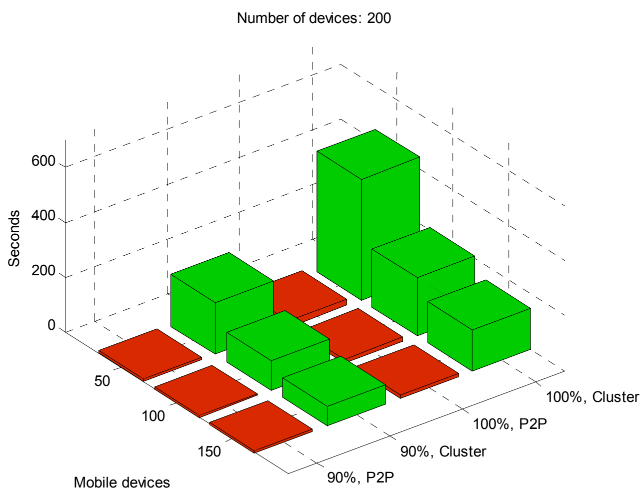

Figure 9. P2P mode is much faster in the denser networks.

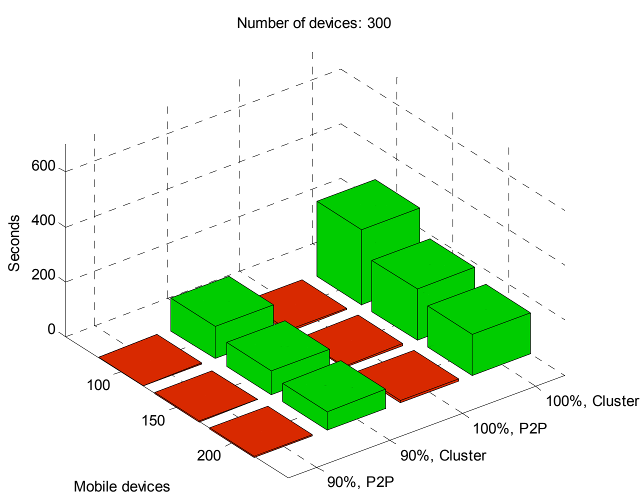

Figure 10. Cluster mode is faster than in the sparser networks.

For 100 devices, P2P mode is about 6 times faster than cluster mode, independently of the number of mobile devices (Figure 8). In a setting with 200 devices, P2P mode is between 25 and 65 times faster than cluster mode, strongly depending on the number of mobile devices (Figure 9). With 300 devices, the difference is a bit smaller, but P2P is still between 30 and 55 times faster than cluster mode (Figure 10).

Nevertheless, even if the time difference of the two modes are much higher in denser networks, the maximum average time needed in cluster mode also decreases from 600 seconds in the sparse network to 300 seconds in the dense network (see Figure 8, 9 and 10).

Generally speaking, P2P is faster by design, because podcast information will be exchanged between any neighboring devices.

In contrast to that, communication in cluster mode is limited to the nodes belonging to the same cluster. Mobility, respectively nodes changing between different clusters, is an essential requirement to spread information across cluster boundaries.

Additionally the clusterhead link stability threshold increases the dissemination time in cluster mode. As the number of devices grows, the characteristics of the topology, in particular network density and mobility degree, gain more impact on the results.

The simulation results show that the information is disseminated faster in networks with a higher number of mobile devices (see Figure 8, 9 and 10).

The situation relaxes considerably when performing similar tests, but aiming at reaching only 90% of the podcast information on each device. The Figures 8, 9 and 10 illustrate the difference in seconds between the average times required in cluster mode compared to P2P mode. It turns out that about half of that time is spent in cluster mode for the dissemination of the final 10% of the podcast information.

6. CONCLUSIONS AND FUTURE WORK

HyCast is a prototypically implemented application designed for podcast dissemination taking advantage of mobile ad-hoc networks. Users can subscribe to podcasts and download episodes both locally within an ad-hoc network as well as using an Internet connection, depending on current connectivity. Such uplinks are supplementary and provide an additional, albeit less preferred, communication opportunity.

Two different strategies for the dissemination of podcast information have been introduced and evaluated. In P2P mode, devices exchange podcast information in an en-passant fashion.

Cluster mode in turn employs WACA, a weighted application-aware clustering algorithm, in order to efficiently structure ad-hoc network partitions. WACA works in a localized fashion, which is a prerequisite with respect to highly dynamic environments where islands of connected devices appear and disappear. In the HyCast application, podcast information is exchanged using the clusterheads elected via.

Roughly speaking, P2P mode induces much more overhead compared to cluster mode. However, P2P results in considerably faster podcast information dissemination than cluster mode. With respect to HyCast, small overhead is much more important than timing issues.

Current work includes exploiting the podcast information and downloading as well as exchanging podcast episodes, i.e.

multimedia data, in an efficient way. The injection-based communication as introduced in the HyMN application will serve as a basis.

In addition to that, we are currently extending HyCast with a map-based visualization of podcasts available in the vicinity. Moreover, we are exploring collaborative filtering approaches to recommend podcasts depending on the user's taste and location.

7. REFERENCES

[1] Aldrich, D., Bell, B. and Batzel, T. Automated podcasting solution expands the boundaries of the classroom *Proceedings of the 34th annual ACM SIGUCCS conference on User services*, ACM Press, Edmonton, Alberta, Canada, 2006.

[2] Andronache, A., Brust, M.R. and Rothkugel, S., Multimedia Content Distribution in Hybrid Wireless using Weighted Clustering. in *2nd ACM Workshop on Wireless Multimedia Networking and Performance Modeling*, (Malaga, Spain, 2006), ACM Press.

[3] Bo, Y. and Ali, R.H., Content-aware search of multimedia data in ad hoc networks. in *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, (Quebec, Canada, 2005), ACM Press, 103-110.

[4] Brust, M.R., Andronache, A. and Rothkugel, S., WACA: A Hierarchical Weighted Clustering Algorithm optimized for Mobile Hybrid Networks. in *The Third International Conference on Wireless and Mobile Communications*, (Guadeloupe, French Caribbean, 2007), IEEE Computer Society Press.

[5] Brust, M.R. and Rothkugel, S., Localized Support for Injection Point Election in Hybrid Networks. in *The Sixth International Conference on Networking (ICN 2007)*, (Martinique, French Caribean, 2007), IEEE Computer Society Press.

[6] Frey, H., Görgen, D., Lehnert, J.K. and Sturm, P., A Java-based uniform workbench for simulating and executing distributed mobile applications. in *Proceedings of FIDJI 2003 International Workshop on scientific engineering of distributed Java applications*, (Luxembourg, 2003).

[7] Görgen, D., Frey, H. and Hutter, C., Information Dissemination based on the en-passant communication pattern. in *Fachtagung 'Kommunikation in Verteilten Systemen' (KiVS 2005)*, (2005).

[8] Lau, W.H.O., Kumar, M. and Svetha, V., A cooperative cache architecture in support of caching multimedia objects in MANETs. in *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, (Atlanta, Georgia, USA, 2002), ACM Press, 56-63.

[9] Malan, D.J. Podcasting computer science E-1 *Proceedinds of the 38th SIGCSE technical symposium on Computer science education*, ACM Press, Covington, Kentucky, USA, 2007.

[10] Nichols, R. and Larraga, L. Using video podcasts to enhance consulting *Proceedings of the 34th annual ACM SIGUCCS conference on User services*, ACM Press, Edmonton, Alberta, Canada, 2006.

[11] Ractham, P. and Zhang, X. Podcasting in academia: a new knowledge management paradigm within academic settings *Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges \& the future*, ACM Press, Claremont, California, USA, 2006.

[12] Rajaraman, R. Topology control and routing in ad hoc networks: a survey. *SIGACT News*, *33*. 60--73.