Cooperative Web Browsing for Mobile Phones

Gian Paolo Perrucci, Frank H.P. Fitzek, Amine Boudali, Maria Canovas Mateos, Peter Nejsum, Sune Studstrup
Department of Electronic Systems
Aalborg University - Denmark
email:{gpp|ff|gr751}@es.aau.dk

Abstract—This paper advocates a novel approach for web browsing on mobile phones based on cooperation among mobile phones within close proximity. In the actual state of the art, mobile phones can access the Web using different cellular technologies such as GPRS, EDGE, UMTS and others. Following the state of the art, higher data rates can only be achieved by increasing complexity, price and energy consumption of the phones. Therefore we propose a novel disruptive architecture where mobile phones are grouped together in clusters using a short range communication such as Bluetooth, sharing their cellular links. In case a mobile phone within the cluster requests a web page, the cellular links of all the devices in the cluster are used to download pieces of the page. When the individual download of partial information is completed, the data is sent to the requesting phone via Bluetooth where the web page is built up and shown to the user. Throughout the paper it will be shown that better performance is achieved in terms of an increased data rate and user experience doing web browsing on mobile phones can be highly improved.

I. INTRODUCTION AND MOTIVATION

Mobile phones of third generation (3G) are dominating the market of cellular systems in these years. Services available on 3G phones are not limited only to voice or Short Message Service (SMS) anymore. The demand for accessing Web services from mobile phone is growing strongly: users want to check emails, news, weather information, share pictures, videos etc. [1]. Therefore phones manufacturers, such as NOKIA [2], have developed web browsers for mobile phones to give a full, desktop-like browsing experience to the users. Even though Internet access from mobile phones is becoming more and more popular all over the world, several users are still reluctant to it. Two of the main reasons are:

1) high prices: Most of the current billing models are still designed for WAP (Wireless Application Protocol) services where small amounts of data were exchanged [3]. However, for those users who want to access web services with their mobile phones, the growth of data traffic leads to a dramatic raise of the costs. To overcome this problem, network operators started new billing models for mobile data, such as flat rates, but prices are still high. Anyhow, the solution to high costs for the connection is out of the scope of this paper.

2) low data rate: Mobile communication poses some problems which do not exist on wired networks, such as limited resources (small displays, battery, etc.) and especially lower data rate. For these reasons, user experience of web browsing on mobile devices cannot be as good as the desktop version. Moreover, accommodating higher data rates on mobile phones increases the complexity of the device as well as the price and the energy consumption.

For example, a way of increasing the date rate is to implement new technologies on the devices such as: Enhanced Data rates for GSM Evolution (EDGE), Universal Mobile Telecommunications System (UMTS), etc. However, the data rate on mobile phones is still low if compared to the one achievable on wired networks and the price of a device supporting these technologies can be too high. As described in [4], wireless devices can benefit from cooperation in multicast scenarios. For example, if two or more terminals within a range of 10-15 meters are downloading the same content over the cellular link, they can decide to cooperate by forming a cluster. Each of them downloads over the cellular link just a portion of the original file and gets the rest via Bluetooth from the others. By doing that, higher data rates are achieved.

In this paper we show an example of cooperation in a unicast scenario where each member of the cluster is interested in a different content, such as a webpage. We propose a novel approach for mobile web browsing that increases the data rate exploiting cooperation among mobile phones. Results show that in a cluster with only two terminals, the speed of a web page downloading is almost doubled. This is a tangible enhancement that users can perceive and therefore their web browsing experience can be highly improved.

The Fig. 1 shows a sample web browsing session. Typically it consists of one or more Web Page Browsing cycles. Each cycle, starts with a Web Page Request Phase and includes other two phases: the Downloading Phase and the Thinking Phase. At the beginning, the session is in a dormant state. After the user makes a request for a web page to the browser (Web Page Request Phase), the Downloading Phase begins. During this phase all data are downloaded and its duration highly depends on the data rate of the cellular link. When the browser displays the requested content on the screen, the Thinking Phase begins. The user is now “using” the information retrieved: for example reading the news, emails etc. This cycle is repeated every time the user requests a new web page. Our aim is to minimize the duration of each Web Page Browsing cycle to enhance the user experience and make it more similar to the desktop one.

However, the duration of the Thinking Phase mostly depends on the user and we cannot force him to reduce it. The duration of the Downloading Phase mostly depends on the data rate provided by the cellular link, instead. We can
reduce it by "virtually" increasing the data rate, exploiting the cooperation among mobile phones. The general concept of cooperative wireless networks advocates to utilize and combine the capabilities of several mobile terminals, instead of considering each mobile terminal as a standalone device. In our case, mobile phones can time-share their cellular link connection during the Thinking Phase in order to increase the virtual downlink data rate.

The rest of the paper is structured as follows: Section II is briefly describing the scenario where the presented approach can be applied. Section III gives a technical description of the architecture of the system. In Section IV results are presented, showing that the data rate for mobile web browsing can be highly increased with our approach. Section V concludes the paper.

II. DESCRIPTION OF THE SCENARIO

Let us consider two mobile phones within a range of 10-15 meters. Let us suppose further that both the terminals have network connectivity capabilities (GPRS, EDGE, UMTS, etc.) and a short range communication air interface, such as Bluetooth. Whenever one terminal wants to initiate a web browsing session, the two phones form a Bluetooth cluster where the first one acts as master and the other one as slave. The master asks the slave to download some of the contents of the webpage (pictures, text, etc.) using the cellular link. As soon as one content has been downloaded, the data are sent to the master via Bluetooth. When all the contents have been collected, the web browser on the master terminal is able to recreate and to show the web page to the user. By cooperating in such a way, the so-called "virtual data rate" of the downloading phase is the sum of the data rates of the two terminals in the cluster. If we do not take into account the time spent to send the last content via Bluetooth (which is small compared to the overall duration of the downloading phase), we can say that the speed of the downloading phase can be doubled. However, the role of the master depends on which phone wants to download the web page. For example, during a Downloading Phase initiated by one phone, can happen that the other one makes a request for a web page. This will cause an overlapping of requests. Therefore, two different scenarios are possible:

1) First scenario where the two Downloading Phases are overlapping with each other;
2) Second scenario where only one terminal at the time is requesting a web page for each web page browsing cycle.

Let us define the virtual capacity as:

\[ C_V = C_{T1} + C_{T2} \]  

where \( C_{T1} \) and \( C_{T2} \) are the cellular link capacity that the master terminal and the slave terminal are dedicating to the cooperation respectively. Let us define the Webpage Browsing Cycle Duration as:

\[ T_{WB} = T_{PR} + T_{DP} + T_{TP} \]  

where \( T_{PR}, T_{DP} \) and \( T_{TP} \) are the Page Request, the Downloading Phase and the Thinking Phase duration respectively.

In the first scenario, the \( C_V \) is the highest possible, because both \( C_{T1} \) and \( C_{T2} \) are dedicated to the cooperation during all the Downloading Phase. In Fig. 2 this scenario is described, by showing the different \( C_V \) in both the cooperative and non-cooperative cases. The orange rectangle represents the \( T_{DP} \). In case of cooperation the \( T_{DP} \) is shorter since the \( C_V \) is higher and consequently the \( C_V \) is shorter if compared to the case without cooperation. The gain achieved by cooperating is represented in the same picture by the red arrows. In the second scenario, both the terminals are in the Downloading
Fig. 3. Downloading Phases overlapping: the Figure shows the virtual capacity in the cooperative and non cooperative web browsing when the Downloading Phases on the two terminals are overlapping with each other. It is increased only when the Downloading Phase of the two terminals are not overlapping. Comparing this Figure with the Fig. 2, you can see that the gain in the cycle duration is smaller due to the fact that the terminals are not cooperating all the time.

Phase at the same time, then each of them is acting selfishly and downloading its own contents (see Fig. 3). However, when not in its own Downloading Phase, the slave offers its cellular link for the cooperation. This means that the $C_V$ is increased only when the Downloading Phase of the two terminals are not overlapping. Therefore, in this scenario, the gain highly depends on the duration of the overlapping and it is smaller, compared to the one in the previous case.

In the real world, the possibility to use the cooperative web browsing highly depends on the number of Bluetooth enabled phones nearby the user that are willing to cooperate. However the number of people that have Bluetooth activated, especially when they are not using it, is questionable. In fact, they tend to have Bluetooth deactivated for two reasons:

- keeping the Bluetooth on, reduces the battery lifetime.
- risk of malware attacks is becoming more and more popular on mobile phones.

In [6] the authors made a measurement campaign to see how many devices, and specifically mobile phones, had Bluetooth activated in public places. Surprisingly, their results show that in such places you can always find several devices with their Bluetooth on.

Moreover, in order to be successful, the cooperative web browsing has to be totally transparent to the user. In fact, most of the users can be annoyed by taking care of the Bluetooth device discovery and cluster forming and it can be very complicated for less experienced users. However, this can lead to security and privacy issues that are out of the scope of this paper.

III. TECHNICAL DESCRIPTION OF THE SYSTEM

In order to evaluate the performance of the cooperative web browsing, a test-bed has been implemented. An application has been developed programming in Symbian C++ which runs on Nokia S60 phones. Specifically two Nokia N70s [5] have been used for testing. The application is installed on both phones. When the web browsing session starts, a Bluetooth connection is established between them, in order to form the cooperation cluster. The master sends a request for the header of the specified web page to the web server and the Web Page Request Phase starts.

After receiving this request, the web server replies to the phone by sending all the information about the requested page. Now the Downloading Phase starts. We can divide this phase in two sub-phases: Web Page Processing Phase and Components Downloading Phase.

A. Web Page Processing Phase

During this phase, the information retrieved during the Web Page Request Phase are processed by the master by performing the following actions:

- The content of the URL is processed.
- A searching process starts looking for a logical predefined pattern in the content of the URL. These patterns represent the paths of all external components in the web page, such as pictures and external links.
- The header of each component and its file size is retrieved. Each component is now associated with its size and put into a list.
- The entries are classified according to their component type, such as an image, an anchor, an external link or an internal link.
- The list is sorted in an ascendant order according to the size of each component.
- The list is split in two sub-lists equally balanced (if possible), based on the component sizes (see Fig. 4).

At the end of this phase, the master phone has two sub-lists containing the links of the components and their sizes. These information are now used to download the components during the next phase.

B. Web Page Components Downloading Phase

When the Web Page Processing Phase is over, the actual download of the components can start:

- The master phone sends one of the two sub-lists of the components to the slave.
- The two phones start to download the components according to the sub-list.
- Both the files, the ones downloaded by the master and the ones downloaded by the slave and transferred via Bluetooth, are stored in a folder on the local hard drive of the master phone.
- As soon as all the contents are downloaded, the web browser builds up the page and shows it to the user.

As explained before, this is the most delicate phase. In fact if the slave requests another web page now, the cooperation
Fig. 4. **Web Page Processing Phase**: this Figure shows how the content of the web page is processed. All the components are classified and inserted into a list according to their size. The list is split into two sub-lists in such a way that the size of them is as much balanced as possible.

![Diagram of web page processing](image)

stops. In this case, the slave sends to the master phone three pieces of information:

1) a sub-list containing the remaining components to be downloaded.
2) the contents eventually downloaded already
3) a request of cooperation as soon as the master ends the **Downloading Phase**

Now the **Downloading Phases** are overlapping and both the master and the slave proceed the download without cooperating. To realize such a system, two different approaches are possible: **Direct web browsing** and **Web browsing using a proxy**. Both of them have the same **Web Page Components Downloading Phase**, but there is a difference regarding the **Web Page Processing Phase**.

1) **Direct web browsing**: In this approach, who takes care of the **Web Page Processing Phase** is the master phone. It contacts directly the web server and requesting the URL header. Afterwards it is taking care of all the information processing and of the two sub-lists generation.

2) **Web browsing using a proxy**: The overall system can be improved by adding a proxy server as an intermediate processing unit between the master phone and the web server (see Fig. 6). The master phone sends the request of the header of the specified web page to the proxy instead of sending it directly to the web server. Upon receiving the request, the proxy establishes a TCP connection to the web server forwarding the request for the header. The proxy is now in charge of the **Web Page Processing Phase** and of the sub-lists generation.

![Diagram of cooperative web browsing with proxy](image)

In this second approach, the proxy is doing the job that before was done by the master phone in the **Web Page Processing Phase**. Therefore, this setup gives better performance in terms of speed and energy consumption compared to the previous one especially if the complexity of the processing phase is high. However, by using the proxy, the system is less flexible and robust. For example, if the proxy is not working...
for any reason, the whole system cannot work.

IV. RESULTS

Several tests have been conducted in order to evaluate the performance of this new system. First of all, the Symbian C++ application has been successfully tested with and without cooperation. Then some measurements have been made to determine the gain achieved by the cooperative web browsing. The measurements consisted of downloading a web page with both the cooperative and non-cooperative approach. It was repeated several times for each approach and for several web pages. When the page was requested, the master phone was starting a timer which was stopped at the end of the Downloading Phase. The durations of all the Downloading Phases were stored in a file and processed afterwards. The measurements were conducted in the Web browsing using a proxy scenario. As it was expected from the theoretical suppositions, there was a gain in the Web Page Browsing Cycle duration by using the cooperative web browsing approach. In average, the duration of the Downloading Phase was reduced by 47% if compared to the non cooperative case (see Fig. 7). This means that the user can open a web page almost twice faster by cooperating!

In this paper, we have proposed a novel approach for web browsing on mobile phones using cooperation. An application for mobile phones has been implemented for the test-bed for evaluating the performance of the proposed system. Results show that the cooperative web browsing (in the Web browsing using a proxy scenario with two terminals in the cluster) can increase the virtual capacity of the cellular link and can reduce the duration of the Downloading Phase by 47%. Therefore this new architecture can be used to overcome the problem of low data rates for Internet access on mobile phones. In fact, cooperative web browsing approach can enrich the user experience of mobile web browsing by significantly increasing the data rate. In contrast with scenarios described in [10], [11] and [12], with this new architecture no benefits are achieved in terms of energy savings. However, comparing cooperative web browsing with the traditional web browsing, the amount of energy spent is almost the same (slightly more if we consider the energy spent for the Bluetooth communication), but the performance is much better. Furthermore, the concept of cooperative web browsing, described above for only two terminals, can be extended to clusters with more terminals. In fact, the more devices are forming the cooperation cluster, the greater the expected gain in data rate is. Cooperative web browsing is applicable, for example, in crowded public areas such as buses, train stations, airports etc. where several users are within a short range.

REFERENCES