

# Practical Issues of Wireless Mobile Devices Usage with Downlink Optimization

Ondrej Krejcar, Dalibor Janckulik, and Leona Motalova

VSB Technical University of Ostrava, Center for Applied Cybernetics,  
Department of measurement and control, 17. Listopadu 15, 70833 Ostrava Poruba,  
Czech Republic  
Ondrej.Krejcar@remoteworld.net, Dalibor.Janckulik@vsb.cz,  
Leona.Motalova@vsb.cz

**Abstract.** Mobile device makers produce tens of new complex mobile devices per year to put users a special mobile device with a possibility to do anything, anywhere, anytime. These devices can operate full scale applications with nearly the same comfort as their desktop equivalents only with several limitations. One of such limitation is insufficient download on wireless connectivity in case of the large multimedia files. Main area of paper is in a possibilities description of solving this problem as well as the test of several new mobile devices along with server interface tests and common software descriptions. New devices have a full scale of wireless connectivity which can be used not only to communication with outer land. Several such possibilities of use are described. Mobile users will have also an online connection to internet all time powered on. Internet is mainly the web pages but the web services use is still accelerate up. The paper deal also with a possibility of maximum user amounts to have a connection at same time to current server type. At last the new kind of database access – Linq technology is compare to ADO.NET in response time meaning.

**Keywords:** Mobile Device; Prebuffering; Response Time; Area Definition, Linq, ADO.NET.

## 1 Introduction

The usage of various mobile wireless technologies and mobile embedded devices has been increasing dramatically every year and would be growing in the following years. This will lead to the rise of new application domains in network-connected PDAs or XDAs (the "X" represents voice and information/data within one device; "Digital Assistant") that provide more or less the same functionality as their desktop application equivalents. The idea of full scale applications pursuable on mobile devices is based on current hi-tech devices with large scale display, large memory capabilities, and wide spectrum of network standards plus embedded GPS module (HTC Touch HD).

Users of these portable devices use them all time in context of their life (e.g. moving, searching, alerting, scheduling, writing, etc.). Context is relevant to the mobile user, because in a mobile environment the context is often very dynamic and the user

interacts differently with the applications on his mobile device when the context is different [1].

My recent research of context-aware computing has been restricted to location-aware computing for mobile applications using a WiFi network (LBS Location Based Services). The information about basic concept and technologies of user localization (such as LBS, Searching for WiFi AP) can be found in my article [2]. On localization basis, We created a special framework called PDPT (Predictive Data Push Technology) which can improve a usage of large data artifacts of mobile devices [3]. We used continual user position information to determine a predictive user position. The data artifacts linked to user predicted position are prebuffered to user mobile device. When user arrives to position which was correctly determined by PDPT Core, the data artifacts are in local memory of PDA. The time to display the artifacts from local memory is much shorter than in case of remotely requested artifact.

The idea of prebuffering may not be only one application method for user position knowledge. As well as WiFi is not only one wireless network to use for localization of user device. WiFi has advantage in speed in indoor positioning therefore the GSM/UMTS can be used in outdoor. The GPS sensor is also embedded in several types of current mobile devices, or it can be plugged by SDIO or BT interface.

We would like to describe a problem of long response in mobile device in the beginning of next chapter. Following subchapter will deal with optimal artifact size determination as well as new partial prebuffering techniques details. The position information obtaining from wireless networks (WiFi, BT, GSM, GPS) background will follow in next subchapter. The needed info about PDPT Framework design, area definition around the user and PDPT client application highlights are in the rest subchapters of chapter 2.

## 2 The PDPT Framework and PDPT Core

The general principle of my simple localization states that if a WiFi-enabled mobile device is close to such a stationary device – Access Point (AP) it may “ask” the provider’s location position by setting up a WiFi connection. If position of the AP is known, the position of mobile device is within a range of this location provider. This range depends on type of WiFi AP. The Cisco APs are used in my test environment at Campus of Technical University of Ostrava. We performed measurements on these APs to get signal strength (SS) characteristics of all APs. The simplification of these characteristics was made to get one “super ideal characteristic” which represent a combination of characteristics of all measured APs. More details can be found in chapter 2.3 [5]. The computed equation for Super-Ideal characteristic is taken as basic equation for PDPT Core to compute the real distance from WiFi SS. This equation is in the web service code (PDPT Framework Server – Core module) to transform a Signal Strength in dB to distance from WiFi APs.

The PDA client will support the application in automatically retrieving location information from nearby location providers, and in interacting with the server. Naturally, this principle can be applied to other wireless technologies like Bluetooth, GSM or WiMAX.

To let a mobile device determine its own position is needed to have a selected adapter still powered on. This fact provides a small limitation of use of mobile devices. The complex test with several types of battery is described in my article [4] in chapter (3). The test results with a possibly to use a PDA with turned on WiFi adapter for a period of about 5 hours.

**2.1 The Need of Predictive Data Push Technology**

PDPT framework is based on a model of location-aware enhancement, which we have used in created system. This technique is useful in framework to increase the real dataflow from wireless access point (server side) to PDA (client side). Primary dataflow is enlarged by data prebuffering. PDPT pushes the data from SQL database to clients PDA to be helpful when user comes at final location which was expected by PDPT Core. The benefit of PDPT consists in time delay reducing needed to display desired artifacts requested by a user from PDA. This delay may vary from a few seconds to number of minutes. Theoretical background and tests were needed to determine an average artifact size for which the PDPT technique is useful. First of all the maximum response time of an application (PDPT Client) for user was needed to be specified.

Nielsen [6] specified the maximum response time for an application to 10 seconds [7]. During this time the user was focused on the application and was willing to wait for an answer. The book is over 20 years old (published in 1994). We suppose the modern user of mobile devices is more impatient so the stated value of 10 second will be shorter. This is for me even better, because my framework is more usable. We used this time period (10 second) to calculate the maximum possible data size of a file transferred from server to client (during this period). To define the amount of data is possible to download to mobile device; we executed a test of data transfer rate measurement of large data size artifacts throw the FTP protocol. we used as test devices four types of PDA (HTC Athena, HTC Universal, HTC Blueangel, HTC Roadster). Fist two devices are equipped with 802.11g standard while the rest two are only with 80211b standard of WiFi capability. These PDA devices were connected throw CISCO Wi-Fi AP. The FTP server holds 3 types of large artifacts (files) which were downloaded to internal PDA memory.

**Table 1.** Transmission speed on large files

	PDA device			
	Athena	Universal	Blueangel	Roadster
<b>data size [MB]</b>	<b>Transfer Speed [kB{s]</b>			
10	347	123	160	106
20	344	121	157	79
40	314	123	58	43

Unafraid the theoretical transfer rates (802.11g = 54 Mbit/s, 802.11b = 11Mbit/s) were not achieved [Tab. 1]. The maximal transfer rate of 350 kB/s has HTC Athena, but this device is not a standard PDA device. Athena is a mini-notebook with windows mobile 6 operating system. All of others devices have only a quarter amount of

such speed. It is much clear that the wireless connected mobile devices have not only a limitation in wireless network module HW, but it has a problem in a slow internal bus. Finally if transfers speed vary from 40 to 160 kB/s the result file size (file which can be downloaded in a defined time of 10 seconds) vary from 400 to 1600 kB.

## 2.2 Determination of Optimal Artifact Size

The next step is an average artifact size definition. we use a network architecture building plan as a sample database, which contained 100 files of average size of 470 kB. The client application can download during the 10 second period from 2 to 3 artifacts. The problem is the long time delay in displaying of artifacts in some original file types. It is needed to use only basic data formats, which can be displayed by PDA natively (bmp, jpg, wav, mpg, etc.) without any additional striking time consumption.

The abysmal difference of transfer speeds from previous table [Tab. 1] vamooses, when we use for transfer smaller data files (10KB – 150KB). The testing of data transfer throw the web services was executed on all of mentioned devices. Firstly the 50 kB and then the 150 kB data file were transferred. The response time for one access and then for 100 access and finally for 500 access were measured. The test results are in [Tab. 2] and [Tab. 3].

**Table 2.** Response time - 50KB artifacts [ms]

	Number of Artifacts		
	1	100	500
Type	Transfer Speed [kB{s]		
Mode	101.82	5208.62	16036.61
Median	101.86	5228.54	16024.24
Average	102.00	5229.81	16022.67

The third step to determine the optimal artifact size is testing of database response for buffering. The test was executed again on all mentioned devices. The information about SQL server response of SQL Server 3.5 Compact Edition was stored.

From the executed test over the web service is evident; the artifacts of size from 50 to 150 kB are more suitable for transfer. It is because the transfer speed of them is relatively affordable in compare to transferred data amount.

**Table 3.** Response time - 150KB artifacts [ms]

	Number of Artifacts		
	1	100	500
Type	Transfer Speed [kB{s]		
Mode	304.82	20208.93	61029.73
Median	301.85	20227.71	61026.08
Average	302.00	20229.99	61022.86

In this case is only a higher starter costs which going to fall after first executed query. The SQL server response results are better in case of 50 to 150 kB artifacts. The artefacts transfer is not striking burdened by dependence on real speed of connection between AP and mobile device. Difference will show after a huge amount of artifacts was transferred (hundreds). The buffer stream of such size is equivalent to download of minimally 14 presentation areas. As will be shown from schema of presenting area, the most of buffered parts will not be needed. The most part of the writing time of SQL buffer from web service is due to the response from SQL server. As it is resulted in tables, writing of tens 50 kB files to the database cost about a same time as writing of one 500 kB file. With smaller files we are able to be more effective to cover buffered area.

**Table 4.** Response time average- SQL query - Insert of 1 artifact into SQL CE database engine [ms]

Device Type	Artifact size[kB]		
	50	150	500
HP 614c	176.2	528.3	2377.3
HTC Roadster	222.4	667.6	3004.0
HTC Advantage	226.6	679.5	3057.6
HTC Blueangel	136.4	409.0	1840.4
Samsung Omnia	185.9	557.2	2507.3

**Table 5.** Response time average- SQL query - Select of 1 artifact into SQL CE database engine [ms]

Device Type	Artifact size[kB]		
	50	150	500
HP 614c	83.0	95.7	118.7
HTC Roadster	91.3	103.3	128.1
HTC Advantage	79.3	91.2	113.1
HTC Blueangel	79.1	91.1	112.9
Samsung Omnia	81.1	93.1	115.5

The large artifacts are better in compare to 1Byte transferred data, but the response time is not optimal.

Such large artifact size is not only with slow transfer speed, but they also allow move with steps, which are not affordable for quicker move or more accurately presentation of position in artifact. The most of tests and problems passes from the size of artifacts. Therefore we made a change to data artifact size to maximal size of 150 kB and we change also the access to these artifacts to new way.

PDPT framework design is based on the most commonly used server-client architecture. To process data the server has online connection to the information system. Technology data are continually saved to SQL Server database.

Previous version of PDPT framework buffered the data according to the predicted position of user. In case of unexpected change of direction, the buffer in most cases cannot act on this quick change.

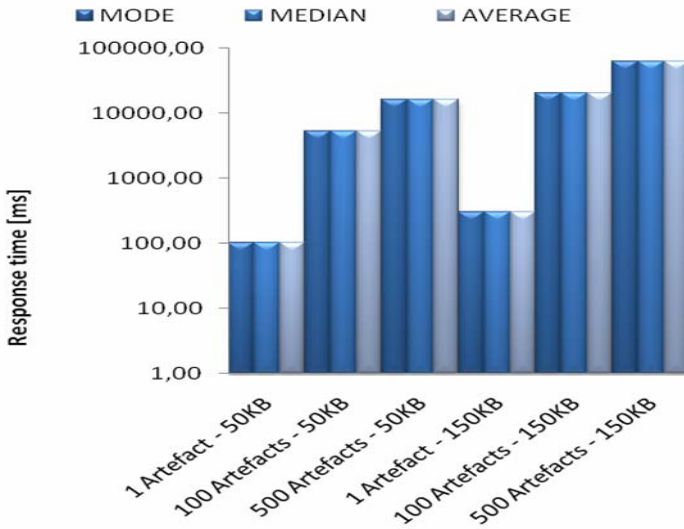


Fig. 1. Web service reaction times

The active presented area was divided to more partial artifacts [Fig. 2b]. This new modified system is now implemented to our other projects, where the position of user is needed. One of these projects is a Guardian II. This project is for hospitals areas for patients and physicians monitoring. We have implemented the new possibilities of biomedical e-health systems are discovered for increasing of interactivity. Based on position of patient, the server can select the nearest physician or nurse to act on discovered problem. By this way the response on problem can be reduced and it can help to save more human life.

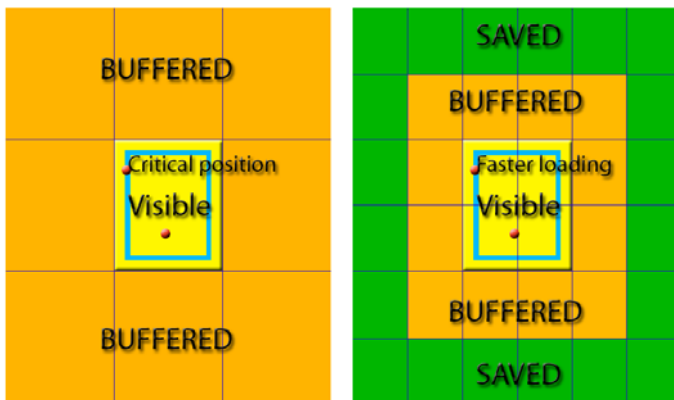


Fig. 2. a) Old application buffering (left fig.). b) New application buffering (right fig.) – visible parts merged to smaller artifacts.

Active area for move of map basis can be not only on outer margin of the whole area, but it can be on borders of several picture-boxes. By this technique is possible to move with map basis in softer grid and allow presenting the actual position of tracked object more precisely. By consequential evaluation of object moving speed, and with suitable modified map basis, we can achieve the effect of zoom of map basis. After application of such principle the system can be applicable for open space with sufficient signal for triangulation - WiFi, GSM or WiMAX. This part of framework is suitable for patient tracking in case of home care agencies. We can track the time of one attendance of nurse at the patient.

### 2.3 WiFi, BT, GSM and GPS Data Collection for Localization

When the use of PDPT technique is possible and it bears some advantages, the first key step of the PDPT is a data collection phase. we record information about the radio signals (WiFi, BT, GSM) as a function of a user's location. The signal information is used to construct and validate models for signal propagation.

If the mobile device knows the position of the stationary device (transmitter, BTS), it also knows that its own position is within a range of this location provider. The typical range vary from 30 to 100 m in WiFi case, respectively 50 m in BT case or 30 km for GSM. Granularity of location can be improved by triangulation of two or more visible APs (Access Points) or BTSs (Base Transceiver Stations).. The PDA client currently supports the application in automatically retrieving location information from nearby WiFi location providers, and in interacting with the PDPT server. Naturally, this principle can be applied to other wireless technologies like BT, GSM, UMTS or WiMAX. The application (locator) is implemented in C# language using the MS Visual Studio .NET with .NET Compact Framework and a special OpenNETCF library enhancement.

In previous research, we focused only to use of WiFi networks while the other wireless possibilities remained without a proper notice. Now we made an enhancement of Locator component of PDPT framework to allow operate with BT and GSM networks.

In BT network case, the position of BT APs must be known to allow the position determination. To collect BT APs position info in outdoor environment, the GPS can be used. For indoor area, the GIS (Geographic Information System) software with buildings map must be used to measure exact position of BT AP against to local environment. To manage with BT hardware of mobile device another library InTheHand 32Feet.NET is used.

GSM network is not local network but a cellular network. The problem is in position information of GSM BTSs. The operator doesn't provide any such information. One of possible solutions is based on unofficial BTSs lists which can be found on internet. The lists are typically available in HTML, TXT or CSV formats. The medium rate for BTSs with GPS position information is about 90 % of all BTSs in European countries. In case of PDPT Framework, the list must be converted to PDPT server database – GSM\_BTS table.

In all three described cases of nearby BSs scanning, the data are saved to Locator Table in PDPT server DB. Data are processed from Locator Table throw the PDPT Core to Position Table. The processing techniques depend on selected wireless

network. WiFi and BT network provide all visible APs nearby the user. From list of these APs is computed actual position (by triangulation).

Mobile devices with windows mobile operation system do not provide any GSM info to .NET Compact Framework. Even any special framework as in previous two cases is not known for me until now. Only possibility is in use of RIL (Radio Interface Layer) library. This library is divided into two separate components, a RIL Driver and a RIL Proxy. The RIL Driver processes radio commands and events. The RIL Proxy performs arbitration between multiple clients for access to the single RIL driver. When a module calls the RIL to get the signal strength, the function call immediately returns a response identifier. The RIL uses the function response callback to convey signal strength information to the module.

The GSM network provide only one BS info in each search cycle. This BS has the highest signal strength. The more BTSs info is collected by a several iteration cycles. During 10 cycles (per 10 seconds) the 4 BTS info is obtained on average.

The important info from BTSs is Signal Strength and Time Advance (TA). SS is refreshed every several seconds (in every scan) whereas TA is provided only during some type of communication with selected BTS (e.g. request to talk, move to another area - Location Area Code (LAC)). The list of these BTSs with info is further processed as in previous case for WiFi and BT networks. Only change is in usage of TA if it is accessible. Another possibility to get the user position in outdoor space is in GPS [8]. GPS provide a position by LONGitude and LATitude (X and Y). Only simple conversion is needed to transform a GPS coordinates to S-JTSK, which is used in PDPT Framework.

## 2.4 The PDPT Framework Design

The PDPT framework design is based on the server-client architecture. The PDPT framework server is created as a web service to act as a bridge between MS SQL Server (other database server eventually) and PDPT PDA Clients.

Client PDA has location sensor component, which continuously sends the information about nearby AP's intensity to the PDPT Framework Core. This component processes the user's location information and it makes a decision to which part of MS SQL Server database needs to be replicated to client's SQL Server CE database [9][10]. The PDPT Core decisions constitute the most important part of PDPT framework, because the kernel must continually compute the position of the user and track, and predict his future movement. After doing this prediction the appropriate data are prebuffered to client's database for the future possible requirements. This data represent artifacts list of PDA buffer imaginary image.

## 2.5 PDPT Core - Area Definition

The PDPT buffering and predictive PDPT buffering principle is not very complicated. Firstly the client must activate the PDPT on PDPT Client. This client creates a list of artifacts (PDA buffer image), which are contained in his mobile SQL Server CE database. Server create own list of artifacts (imaginary image of PDA buffer) based on area definition for actual user position and compare it with real PDA buffer image.



The area can be defined as an object where the user position is in the object centre. The PDPT Core will continue with comparing of both images. In case of some difference, the rest artifacts are prebuffered to PDA buffer. When all artifacts for current user position are in PDA buffer, there is no difference between images. In such case the PDPT Core is going to make a predicted user position. On base of this new user position it makes a new predictive enlarged imaginary image of PDA buffer. The new cuboid has a center in direction of predicted user moving and includes a cuboid area for current user position. The PDPT Core compares the both new images (imaginary and real PDA buffer) and it will continue with buffering of artifacts until they are same. In real case of usage is better to create an algorithm to dynamic area definition to adapt a system to user needs more flexible in real time. For additional info please refer to [11].

**2.6 Results – Comparisons of Two Techniques for Database Access**

The PDPT framework consists of two main parts – server and client. Supposed number of users is from tens to hundreds of people with mobile device. The PDPT server has a web service interface which must be tested on adequate response time to user requests. We created special software to test these web services interfaces. The results are in table [Tab. 6].

**Table 6.** Median and mean response times for hundreds of users – PDPT Server web services [ms]

	2x4GHz + 6GB DDR2 RAM (Server)		AMD 850MHz + 512MB SD RAM (PC)	
<b>Users</b>	Median	Mean	Median	Mean
200	388	387	1413	1413
300	590	599	2099	2103
400	826	830	2851	2850
500	1061	1037	3516	3508
650	1182	1258	4574	4565
800	1662	1616	5645	5649
1000	1797	1977	7050	7046

Graph at [Fig. 3] describe the increased response time with a increased number of users. As it can be seen the values of several columns varies with small difference even for both server stations (named „Server“ and „PC“). These are unfortunately only small difference. The difference between the stations increased largely from 500 simultaneous users (the difference between the stations is 2.5 s) and is growing up to 5 seconds for 1000 of users. The response time of old PC station is in this case about 7 second. This time is near the maximum response time declared by Nielsen and Hacklay [6][7]. The use of such server station is not suggested and only the new server station can manage with such amount of online users. Our created software is unfortunately possible to evaluate every web service (even the new ones) and before a real usage we can suggest a minimal server hardware configuration for adequate usage.

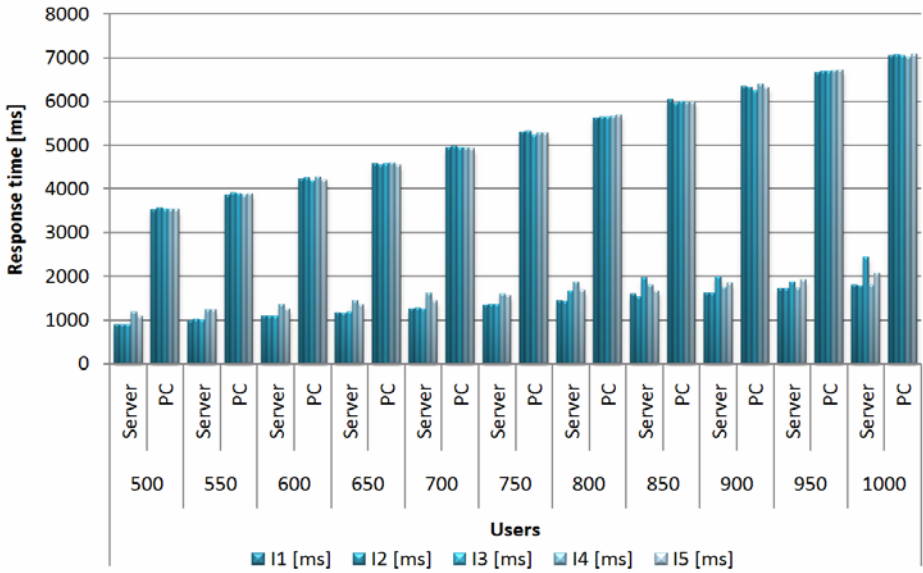


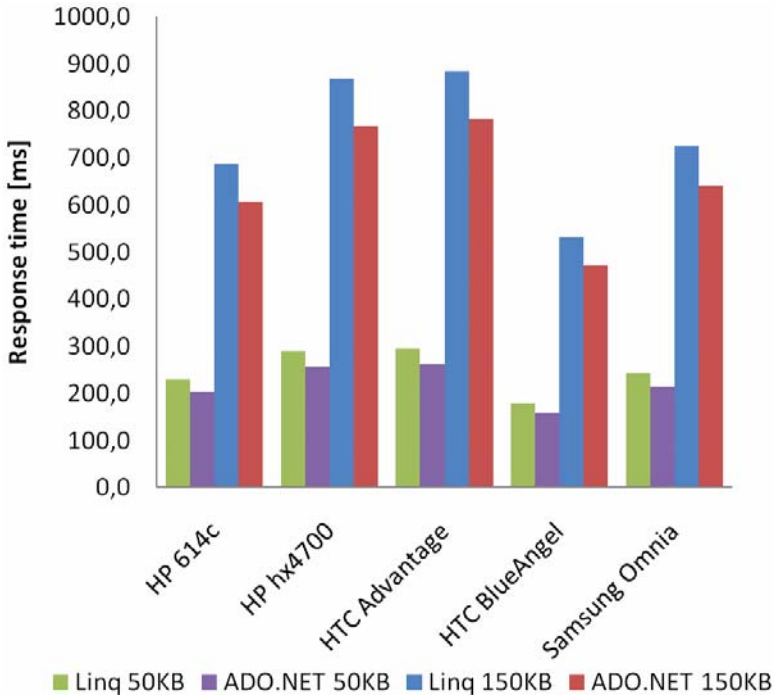
Fig. 3. Graph of response times for accessing of PDPT Server web services – 5 iteration and two server stations were tested

Table 7. Average response time – Insert of one 50KB and 150 kB artefact into SQL database through different technologies [ms]

Artifact size	50 kB		150 kB	
Device Type	Linq	ADO.NET	Linq	ADO.NET
HP 614c	229.1	202.6	686.8	607.5
HTC Roadster	289.2	255.8	867.8	767.7
HTC Advantage	294.6	260.6	883.3	781.4
HTC Blueangel	177.4	156.9	531.7	470.3
Samsung Omnia	241.6	213.7	724.3	640.8

The next step of making results is in mobile client testing. Namely the database access techniques and its response times. We made a test to accessing of SQL server buffer and the selection of better one was made. In present the testing of technologies like LINQ, ADO.NET and the direct access using SQL queries is being realized.

Preliminary test are graphed in figure [fig. 4] to provide a more visual summary. In all cases the Linq access reach higher response time value than the ADO.NET technique. The question is what the Linq technology can bring to user or developer. We evaluate several such things to find a decision which techniques is better to use. Of course the Linq is 13 % slower than the ADO.NET, but we found a several contributions in Linq to surpass this 13 %. Final decision is to use Linq access and technology in all cases which need the perfect visual response to user and access to more than one database table. These two main benefits can help to develop and use the new kind of mobile clients. PDPT Client will take advantage of these benefits in next version which is currently under the development.



**Fig. 4.** Graph of average response times for inserting artifacts trough different technologies on a few devices

### 3 Conclusions

We are focused on the real usage of the developed PDPT Framework on a wide range of wireless connected mobile devices and its main issue at increased data transfer. For testing purpose, five mobile devices were selected with a variety of hardware and software capabilities. The high success rate found in the test data surpassed our expectations. This rate varies from 84 to 96 %. Please see the chapter 4 [5] for more info. The PDPT prebuffering techniques can improve the using of medium or large artifacts (from 50kB to 500 kB per artifact) on wireless mobile devices connected to information system. The localization part of PDPT framework is currently used in another project of biotelemetrical system for home care named Guardian II to make a patient’s life safer [12]. Another utilization of PDPT consists in use of others wireless networks like BT, GSM/UMTS, WiMAX, or in GPS. This idea can be used inside the information systems like botanical or zoological gardens where the GPS navigation can be used in outdoor. Some improvements of Locator module are also described. The larger area of PDPT utilization can improve importance of PDPT Framework in wireless mobile systems.

**Acknowledgment.** This work was supported by the Ministry of Education of the Czech Republic under Project 1M0567.

## References

1. Abowd, G., Dey, A., Brown, P., et al.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707. Springer, Heidelberg (1999)
2. Krejcar, O.: User Localization for Intelligent Crisis Management. In: AIAI 2006, 3rd IFIP Conference on Artificial Intelligence Applications and Innovation, Boston, USA, pp. 221–227 (2006)
3. Krejcar, O., Cernohorsky, J.: Database Prebuffering as a Way to Create a Mobile Control and Information System with Better Response Time. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 489–498. Springer, Heidelberg (2008)
4. Krejcar, O.: PDPT Framework - Building Information System with Wireless Connected Mobile Devices. In: ICINCO 2006, 3rd International Conference on Informatics in Control, Automation and Robotics, Setubal, Portugal, August 01-05, 2006, pp. 162–167 (2006)
5. Krejcar, O., Cernohorsky, J.: New Possibilities of Intelligent Crisis Management by Large Multimedia Artifacts Prebuffering. In: Ulieru, M., Palensky, P., Doursat, R. (eds.) I.T. Revolutions 2008. LNICST, vol. 11, pp. 44–59. Springer, Heidelberg (2008)
6. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
7. Haklay, M., Zafiri, A.: Usability engineering for GIS: learning from a screenshot. *The Cartographic Journal* 45(2), 87–97 (2008)
8. Evennou, F., Marx, F.: Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing* (2006)
9. Arikan, E., Jenq, J.: Microsoft SQL Server interface for mobile devices. In: Proceedings of the 4th International Conference on Cybernetics and Information Technologies, Systems and Applications/5th Int. Conf. on Computing, Communications and Control Technologies, Orlando, FL, USA, July 12-15 (2007)
10. Jewett, M., Lasker, S., Swigart, S.: SQL server everywhere: Just another database? Developer focused from start to finish. *DR DOBBS Journal* 31(12) (2006)
11. Krejcar, O.: Utilization Possibilities of Area Definition in User Space for User-Centric Pervasive-Adaptive Systems. In: Hesselman, C., Giannelli, C. (eds.) Mobilware 2009 Workshops. LNICST, vol. 12, pp. 124–130. Springer, Heidelberg (2009)
12. Janckulik, D., Krejcar, O., Martinovic, J.: Personal Telemetric System – Guardian. In: Biodevices 2008, Insticet Setubal, Funchal, Portugal, pp. 170-173 (2008)