

## **Why test applications for wireless networks?**

*"Doesn't software work, if you want it to work?"*

November 2004

## Contents

Abstract .....	1
Overview .....	1
Problems .....	1
Varying capacity.....	1
Latency and Jitter .....	2
Data corruption .....	2
Temporary interruptions.....	2
Network Address Translation.....	2
Unpredictability.....	2
Protocols .....	3
TCP.....	3
SIP.....	3
Multi-user systems .....	3
Streaming .....	4
Solutions.....	4
Testing and tuning.....	4
Environment.....	4
TCP.....	4
Other protocols.....	5
TCP replacements .....	5
SCTP.....	5
Validitas CAIS.....	5
Conclusions .....	5
References.....	6

# Why test applications for wireless networks?

*Doesn't software work, if you want it to work?*

Erkka Sutinen

Chief Engineer, Validitas Ltd.

## Abstract

Wired and wireless networks differ fundamentally in their behaviour. For this reason, protocols and applications designed either for or on wired networks do not work on wireless networks as intended. As a result, product rollout to wireless networks may fail. Good examples of problem areas are streaming media applications and multi-player games. Current testing environments do not allow comprehensive testing of applications before rollout. However, the newly introduced Validitas CAIS provides a platform that enables extensive testing of mobile applications and devices.

## Overview

The question in the title of this paper is highly valid. For most applications and software projects, the question of testing is raised, but in reality ignored. Timetable pressures and increasing costs usually mean that products are rushed to users without complete testing, which effectively leaves testing to the users. What testing is carried out is based on the assumption that everything that the programming crew has tested, will also work for the end users.

Enter the wireless networks. Their applications and protocols are often developed on standard workstations and servers over local area networks. The problem with this approach is that wireless networks behave completely different than, for example, the Ethernet. The notion that protocols behave in the same way in, say GPRS and the Ethernet, is not correct. Such basic protocols as TCP (Transmission Control Protocol) of the TCP/IP frame have severe problems in GPRS, not to mention more complicated systems such as those involving streaming media.

The issues discussed in this document are valid for all wireless networks, but the main emphasis here is on GPRS (General Packet Radio Service) and WLAN (Wireless Local Area Network) networks.

## Problems

Wireless networks differ from wired networks in several ways. This means that applications and protocols that are fully functional in wired networks do not work as efficiently in wireless networks if they work at all. As a consequence applications developed and tested in wired networks may fail their rollout to wireless networks.

### Varying capacity

The assumptions made in the context of wired networks differ from those in wireless networks. For example, in wired networks it is valid to assume that packet losses are caused of a congested network, so the problem can be solved with a congestion control algorithm, which reduces the speed of the connection. In wireless networks, however, capacity loss is a much more frequently encountered problem. The air interface connection can be lost, for example, when a mobile phone moves to a position where radio waves cannot be transmitted to or where waves reflected from several objects cancel out each other. A case in point is going through a tunnel. Also the network itself can cause problems such as GPRS cell updates, which occur when a mobile phone moves from an area handled by one base station to that of another. The result is a loss of capacity, which can last from a few microseconds to several seconds.

Most protocols used over this type of link use

congestion control methods with some type of congestion avoidance algorithm, which incorrectly sees the network as congested and reduces the transmitting speed. Consequently when the network is good again, the connection is not running at full capacity. Thus in reality GPRS, for example, fails to deliver the 20 kbits/sec or whatever the capacity might be but rather less.

### **Latency and Jitter**

Latency is the time a message takes to travel through a system, for example from a content server to a mobile phone. Jitter is the variation of latency.

In wired networks, like DSL connections, latency delays are usually small and jitter nonexistent. Thus, any changes in speed, delay or latency can safely be assumed to be due to congestion or some other problem, and protocols usually act defensively, lowering transmission speed to avoid congestion.

In wireless networks, however, long delays and variations in latency are default behavior. In mobile packet-switched networks, for example in GPRS, the latency varies a lot and can be up to several seconds. Protocols written for wired connections interpret the situation often incorrectly.

To use the network efficiently, applications should assume that the latency and capacity may go up as well as down at any moment. Because of this, tuning protocols and applications requires a lot of testing in different situations.

### **Data corruption**

Wired networks rarely corrupt data packets. In wireless networks, however, corruption is a common occurrence depending on communication parameters. For example, in GPRS, the Quality of Service (QoS) profile defines whether the radio communication part itself resends corrupted data.

Why is this important? Simply because better QoS parameters may not be available for all

users, or users may not set them and even if they do, their rates are higher. It is true that transport layer protocols like TCP and UDP (User Datagram Protocol) do check, whether data is corrupted or not. But since they cannot fetch only the corrupted parts, they throw away entire datagrams. This means that even if a small part of large packet is corrupted, the whole packet is dropped. As a result, even uncorrupted data is resent. If the air interface condition deteriorates further the protocol stack will drop even the data that gets through the air interface properly.

### **Temporary interruptions**

In wired networks, complete loss of service is rare and users can be expected to restart whatever they were doing. This might involve reloading the web page they were receiving or re-logging into a service.

Wireless networks, on the other hand, are afflicted with frequent service interruptions. What is worse, even if the system automatically re-connects to the network, each time it does so it may utilize a different IP address (which is typical for GPRS connections, for instance). Subsequently, users are often required to restart their applications or re-login to the service.

### **Network Address Translation**

Nowadays, it is common for wired networks to use NAT (Network Address Translation) and wireless networks use it almost without exception. The problem with NAT is that the computer, which connects to a particular service, does not know what address the computer providing the service sees the connection as coming from. Many problems discussed above could be avoided if UDP were used instead of TCP, but as UDP is connectionless, it usually requires application-aware firewalls to ensure that all data go through the NAT system.

### **Unpredictability**

The main problem with wireless networks is that you cannot foresee what the network is going to do. To give an example, if you test a

GPRS application in a real network in the office, you can only be assured that the application works in that network in that office. When you move the situation may change dramatically.

If the user of some mobile application or game is moving in cellular networks, like GPRS or WLAN, the networks occasionally change cells. Each cell change is accompanied by a temporary loss of service. In addition, moving causes delays and capacity changes. While running the application, these situations can be tested by walking or driving around. Unfortunately the results do not have wider applicability, since this is a non-repeatable process. Moreover, events that can be foreseen but are difficult to force, like GPRS routing area updates, are almost impossible to test in real networks and often require special procedures with the network. As a result these events cannot be fully tested in live networks.

## Protocols

### TCP

TCP (Transport Control Protocol) is the most-used transport-layer protocol within the TCP/IP protocol suite. TCP is employed in a wide range of applications including WWW, email and file transfer applications. The use of several algorithms to enhance connections makes TCP rather complicated but, on the other hand, it is also well researched. The TCP/IP stack currently allows a large number of options and parameters. Most programs, nevertheless, do not tune the network connection. Instead they rely on default parameters, which are designed primarily for wired networks. Various experiments, such as those reported by [Leang], have shown that the absence of optimisation makes the performance of TCP less than ideal.

One source of difficulty is the algorithm used for congestion control. This algorithm tries to prevent congestion in the network by observing lost packets. If many packets are lost, it assumes that the network is congested and decreases the datagram-sending rate. This is a perfectly valid assumption in a wired network. If all users

observe this policy, the core network will not get congested and no capacity is lost. In wireless networks, however, packet losses are not caused by congestion, but by temporary connection losses. As the congestion control algorithm perceives these losses as indicating reduced capacity, it does not allow full use of available capacity.

Also the error recovery algorithm can cause problems, when it tries to resend lost packets before they are lost. In wired high-capacity networks this improves connection speed, but in wireless networks it produces unnecessary traffic, which can disturb the data flow, particularly when using the full capacity of the connection.

Flow control may also decrease available capacity by measuring the round-trip time (RTT) without taking into account that in wireless networks only a small percentage of packets have unusually long RTTs.

See [Balak] and [Leang] for more information.

As for other problems caused by TCP-based protocols, like HTTP, see [Chak].

### SIP

Session Initiation Protocol (SIP) is used in Voice over IP (VoIP), streaming Internet radio and, in the future, in the Internet Multimedia Subsystem (IMS), which are included in newer UMTS releases. The biggest problem with SIP in wireless networks today is NAT. Making protocols NAT-friendly is simple: just follow the guidelines presented in [RFC3235], including "Don't put IP addresses in protocol", or "Client originates all connections". Unfortunately, SIP is older than these guidelines, the consequence being that current standard implementations of SIP do not work over NATted links.

### Multi-user systems

Good examples of problematic multi-user systems are multiplayer games. The server managing the game experiences different problems in wireless than in wired networks.

The first problem is that each connection has a different speed. Testing a game in a LAN is not a problem, since each connection is equally fast. But the GPRS connections speed typically ranges from zero to a few tens of kilobits per second. This means that connections are asynchronous and also that, when TCP is used, writes will be blocked from time to time. If normal blocking I/O is used, the server may not be able to write the whole block of data it was trying, resulting the entire server process to freeze as it waits for one receiver to be able to read more.

Moreover, in GPRS and other wireless networks, connections are occasionally lost. Should the game forget the session every time this happens, the players will not be getting their money's worth. An additional problem with lost connections is that users may receive a different IP address for every connection. If a player is required to identify himself after each connection loss, the game is in practice unplayable. These same problems apply to any multi-user system, including mail or calendar services.

### **Streaming**

Sending continuous data, such as voice or game situation event over a wireless network is also problematic. Rather than requiring that every single packet gets through, streaming applications like Voice over IP (VoIP) demand that delays to be constant without too much jitter. The problem is that, since delays are unpredictable, protocols tuned for wired networks tend to get over-protective and increase buffering delays until the system becomes unusable. The solution offered by [Bell] is to carefully select the right kind of codec and to require a good quality of service for the connection - provided that the required QoS is available. Further problems for some protocols are caused by NAT, which needs to be aware of the protocol used.

### **Solutions**

The main problem is that, although wireless networks are nothing new, their impact on communication has only recently attracted the

research it deserves and requires. As a result there are no off-the-shelf solutions currently available.

### **Testing and tuning**

To ensure that the application works in its intended environment you have to test it in that particular environment. Unfortunately testing an application under the circumstances prevailing in the office is not enough, as you cannot force problems into the network. Consequently you are unable to emulate all situations that the users will encounter.

### **Environment**

You can test your application in a LAN, but that eliminates the effects of proper delays. Thus you have no way of knowing whether the application is going to work in a live network. Alternatively, you can test the application in a real network, but that only tells you that it works in that specific network under a particular set of circumstances. What you cannot find out for sure is if it's going to work in a different network with a different setup. Nor you can find out if it's going to work with a heavy load on a congested network.

A far better solution is to test the application on a simulated network environment, using different network air interface conditions and different network setups. Further, a simulation also allows you to explore otherwise un-testable network problem situations.

### **TCP**

Most TCP/IP implementations allow changing the parameters of the TCP protocol. One good idea is to turn off the Nagle algorithm, which tries to pack several smaller packets into one big data packet. More useful hints can be found in [Tier], [Eures] and [Buff]. This type of tuning is dependent on the operating system; in Linux, for example, you can either set the parameters of the entire system, or set the behaviour of each TCP socket separately using `setsockopt(2)` or `ioctl(2)`.

## Other protocols

TCP and UDP are not the only solutions. If tuning the TCP proves inadequate and UDP cannot be used, remember that all IP networks carry any transport layer protocol over the network without modifications. All that is required is that both your server and the client systems understand this protocol.

## TCP replacements

There are several protocols that have been designed as wireless-aware replacements of TCP. You can find more information about these protocols including I-TCP, MTCP, METP and WTCP, on the Internet or from the article by [Anna], for instance. Also SNOOP and TULIP are worth of studying.

## SCTP

The most likely new protocol to be used along TCP and UDP seems to be the Stream Control Transmission Protocol (SCTP). Mostly because it has been standardized into the UMTS release 6 network. SCTP combines a connection-oriented model with several data streams over one connection. Consequently, once a connection has been formed between two hosts they can use several data streams of different characteristics to communicate with each other. In this system, the designer of the application can choose the most suitable communication mode for each use.

## Validitas CAIS

Validitas CAIS is a wireless network simulator, which can simulate any network type and their

common problems. It can be configured to provide the various data delay, jitter, throughput, data corruption and other characteristics of currently available and planned wireless networks, including e.g. GPRS, UMTS, GSM, and many others.

Validitas CAIS also has the capacity to create problems for the connection when triggered. These problems include cell updates and capacity changes that occur, for example, when driving through a tunnel. In addition, it can also provide the same interfaces to the application server as a real network, produce charging tickets (CDRs) and provide subscriber identity information, such as an IMSI or a phone number.

## Conclusions

Currently, most applications are not tested well enough. They are developed in some environment and then published into a real network, in the hope that they would work. As a result, many service rollouts have been less than successful, as the products have been debugged and fixed with real customers.

Testing all foreseeable situations and problems, before the commercial launch of a service saves money, since the application does not acquire a bad reputation due to initial, undetected, bugs. The best available way of testing against the problems described above is Validitas CAIS.

## References

[Anna] Narayanan Annamalai, Lokesh Balakrishnan, Vinod Kumarasamy: TCP for Wireless Networks.

[Balak] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, Randy H. Katz: A Comparison of Mechanisms for Improving TCP Performance over Wireless Links (1996) IEEE/ACM Transactions on Networking

[Bell] Boris Bellalta, Miquel Oliver, David Rincn: Performance Of The Gprs Rlc/Mac Protocols With Voip Traffic

[Buff] Robert Buff, Arthur Goldberg, Web Servers Should Turn Off Nagle to Avoid Unnecessary 200 ms Delays

[Chak] Rajiv Chakravorty, Andrew Clark, Ian Pratt: GPRSWeb: Optimizing the Web for GPRS Links.

[Eures] Eurescom project P605 deliverable: A Guide to Implementing Broadband Multimedia Services.

[Leang] Leang Tzeh Yeu, Joshua Liew, Winston K.G.Seah: Experimentation of TCP schemes over GPRS & WLAN, proceedings of Fourth IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Sep 9 - 11, Stockholm, Sweden.

[RFC3235] RFC 3235: Network Address Translator (NAT)-Friendly Application Design Guidelines

[Tier] Tierney, Brian L: TCP tuning guide for distributed applications on wide area networks. ;login: February 2001.



### Validitas Ltd

Hallituskatu 13-17 D  
FIN-90100 Oulu  
Finland  
phone +358 8 3118 888  
e-mail: info@validitas.com

### Validitas APAC Ltd

Suite 1002, Chuang's Tower  
30-32 Connaught Road Central  
Hong Kong  
phone +852 3421 2032  
e-mail: info@validitas.com

Copyright © 2004, Validitas Ltd. All rights reserved. Validitas and Validitas CAIS are trademarks or registered trademarks of Validitas Ltd. All other trademarks are property of their respective owners. Validitas assumes no responsibility for any errors or inaccuracies in this document. Validitas reserves the right to change, modify or otherwise revise this publication without notice.