Six in the City: Introducing Real Tournament – A Mobile IPv6 Based Context-Aware Multiplayer Game

Keith Mitchell, Duncan McCaffery, George Metaxas and Joe Finney
Distributed Multimedia Research Group
Lancaster University, Lancaster, U.K. LA1 4YR
Tel: +44 1524 65201. Fax: +44 1524 593608
{km, mccaffed, metaxas, joe}@comp.lancs.ac.uk

ABSTRACT

It is rapidly becoming clear that entertainment will be one of the killer applications of future wireless networks. More specifically mobile gaming is predicted to be worth $1.2 billion by the year 2006 to providers in the U.S. alone [26]. The driving force behind this is the introduction of powerful feature rich handsets and ubiquitous access to high performance wireless networks. However, mobile applications face issues that are subtly different from fixed network applications, including fluctuating connectivity, network QoS and host mobility issues. To investigate the requirements of future mobile applications we have deployed a wireless MAN consisting of GPRS and 802.11 hotspots based on Mobile IPv6 around the city of Lancaster and have built an augmented reality game designed to evaluate future mobile application requirements.

In this paper we introduce Real Tournament, our prototype multi-player mobile game, which uses handheld computers augmented with an array of sensors to enable true mobile interaction in a real-world environment. We then evaluate current approaches to real-time interaction and follow by outlining our own architecture more suited to wireless environments and based on the peer-to-peer approach. The approach provides adaptation, shared state, and consistency mechanisms in order to provide support for scalable, low latency, soft real time mobile applications.

Keywords

Mobile IPv6, Context-Aware, Ubiquitous, Gaming, Multimedia, Wireless overlay networks.

1 INTRODUCTION

The Mobile IPv6 Systems Research Lab (MSRL) is a new research collaboration between Cisco Systems, Microsoft Research, Orange and Lancaster University [18]. Together with our industrial partners, local citizens, businesses and authorities, we are in the process of deploying an infrastructure designed to facilitate ubiquitous access to information and services throughout the University campus, city and the surrounding area. The project aims to harness the knowledge and experiences gained from previous research projects such as GUIDE [5,6], LandMARC [16,20] and more recent the Smart-Its project [13]. MSRL primarily involves work in three major areas:

Infrastructure: The deployment of a metropolitan area wireless overlay IPv6 based network is an ongoing area of work within the overall project.

Middleware: The development of a series of abstractions and software components to support the rapid prototyping and evaluation of higher level services and user applications.

Applications: The design, development and deployment of applications which exploit the underlying wireless infrastructure and middleware support to enable novel mobile context-aware applications [2,5] and intelligent multimedia services to be trialed.

The main objectives of the collaborative venture are:

- The provision of a fully operational Mobile IPv6 networked environment.
- To trial new mobile services on users ranging from the University campus, the city, and the region.
- To support research into the general field of mobile computing including, specifically host mobility in wireless overlay networks, multimedia content distribution to mobile

devices, context and location aware ubiquitous services for users on the move and, security and privacy in pervasive computing environments.

2 BACKGROUND AND MOTIVATION

The Mobile IPv6 Systems Research Laboratory has been designed to extend and build upon our previous success within the areas of mobile and ubiquitous computing [25]. More specifically, we feel that the research we have carried out within these individual areas are becoming increasingly tightly coupled, and that issues pertaining to each must be explored simultaneously. Furthermore, we consider two concepts to be vital to the success of future mobile systems, *host mobility* and *context-awareness*.

2.1 Host Mobility

The LandMARC project was carried out in collaboration with Microsoft Research, Cambridge U.K. One of the main outcomes of this collaborative venture was an implementation of Mobile IPv6 for the Microsoft Windows operating system to support research into mobile networks and communications. Mobile IPv6 provides location transparency for mobile devices while roaming on different layer-3 networks. It allows a mobile device to retain the use of its "home address" after moving from the home network to a foreign network or when roaming between foreign networks, so that applications wanting to contact the device at that address may do so independent of its actual location. Mobile IPv6 does this in a way such that network connections bound to the home address of a mobile node survive movement of the device and such that communication takes place directly between the peers.

Since a number of our current research projects aim to explore the various aspects of network mobility, the project focused on the development of Mobile IPv6 for Windows CE during the final months. In close collaboration with the Microsoft CE development team, we ported our Mobile IPv6 stack for Windows to the Windows CE 3.0, CE 4.1 and Pocket PC platforms. The availability of Mobile IPv6 on Pocket PC allows us to use off-the-shelf handheld devices for our experiments and user trials.

2.2 Context Capture and Awareness

The valuable insights and experiences gained as part of the GUIDE initiative [12] have provided us with an initial metropolitan area wireless infrastructure on which to extend and expand our latest test-network. The initial wireless infrastructure was used to experiment with a mobile context-aware tourist guide for visitors to the city.

The aim of the project was to investigate the provision of context-aware mobile multimedia computing support for city visitors. More specifically, the project aim was to develop hand-portable multimedia end-systems which were designed to be context-aware, i.e. have knowledge of their users and their environment including, most importantly, their physical location and their personal preferences. This information was then used to tailor system behavior in order to provide users with services such as context based retrieval of a large hypertext information base and the generation of dynamic guided tours of the city. These tours were created based on a wide variety of factors including location, weather forecast, available budget and time of day [7].

3 REQUIREMENTS

In order to investigate the salient challenges in developing for future context sensitive applications, we chose a single application, and developed it from conception through implementation to evaluation. We then use the experiences gathered to draw conclusions on the demands of such applications, and to discover effective ways to meet those demands. To derive a suitably challenging application, we began by defining high-level requirements for the trial application. These are as follows:

- **Investigate Future Applications** – We believe that future mobile systems will be able to incorporate many more facets of the user's operating environment than we see today, as embedded sensors become cheaper and more widely deployed. Therefore we aim to develop a realistic and complex mobile computing application.
- **Produce Test Data for Underlying Networks**– Since the wireless overlay network is based on Mobile IPv6, we aim to use the mobile applications developed to place realistic stress on the underlying infrastructure. We must therefore make use of a range of different classes of network traffic, e.g. reliable transfer, real-time, high bandwidth communications and low latency networking, in order to adequately evaluate the performance of Mobile IPv6 within future wireless environments.

Using these requirements as a guideline we began to search for the 'killer application'. Moreover, we attempted to identify an area of growth within the mobile market place in order to develop a suitable demonstrator. We examined trends within the current mobile telephony marketplace and identified entertainment as an area of ongoing growth. More specifically, *mobile gaming*.

The capabilities of mobile terminals are increasing at a phenomenal rate (in particular cellular handsets such as the Windows Powered Smartphone 2002 platform, which has the power to execute extremely resource hungry programs – such as Eidos' Tomb Raider application as demonstrated on a cell phone at MDC 2002 [17]).

Most mobile gaming applications are as yet single user, stand-alone applications, or non-real time games based on SMS text messaging. However, given the proliferation of fixed network online multi-player gaming we believe it is only a matter of time before applications become prolific on mobile terminals too.

Within the current telecoms industry there are a number of ongoing initiatives such as downloadable games consisting of several levels, the creation of more-advanced games terminals (such as the Nokia 9210 Communicator) and a commitment to a new range of devices in the entertainment category, such as the Nokia 5510.

It is expected that the penetration of mobile services, together with the revenue focus on data, will ensure that most handheld telecom devices will be able to support interactive games which require real-time communication with remote services in addition to those embedded within the device [21]. Many features of the existing fixed gaming industry will therefore transfer to the business of providing mobile games. It has been predicted that by the end of 2003, 80 percent of mobile telecom devices will have the capability to support server-based or downloadable games, a market which is expected to grow to cover nearly 200 million consumers and 6 billion US dollars by 2005 [21]. We therefore see an enormous potential in value added services such as multi-player games beyond the current trend towards downloadable games.

In striving to look for a suitable complex mobile, real time, distributed application making use of a wide variety of contextual facets, we found the concept of a multi-player, mobile computer game extremely applicable and timely.

4 REAL TOURNAMENT

We are now in the process of developing a multi-player mobile gaming application which we call *Real Tournament (RT)*[1]. Real Tournament is based around handheld multimedia end systems (currently Pocket PC 2002 based iPAQs). Each game player has one mobile device and the game has been designed to be played by teams of approximately four players aged 10 years or above.

Real Tournament is an *augmented reality* multiplayer mobile game – a combination of both real world and virtual reality components. Real world artifacts, such as players, become intrinsic elements within the game's narrative. For example, the context of each player, e.g. their physical location, orientation and player status, become key factors in the game play as this information is used to augment the virtual-reality environment of each and every player. In order to obtain player context, we have augmented each player's mobile device with an array of sensors, including a GPS device and an electronic compass. Therefore, a wide range of personal contexts can be obtained for each player and this can be used to drive the game play (such as their location within a game arena). Furthermore, there is a complex interaction between the physical and virtual entities in RT, which are used to trigger a wide range of context-based game events (which are further described later in this paper) [3]. RT uses a Mobile IPv6 enabled wireless network to share game state in real-time between players.

4.1 Real Tournament Game Arena

The game arena for Real Tournament is a public park (Williamson Park) close to the city centre. The park has at its centre a memorial (known as the Ashton Memorial), a butterfly house and mini-beasts museum (a museum housing creatures such as reptiles, insects and birds). The wireless access points are located in the memorial, which also acts as the starting point for the game.

The narrative for the game relates to a mad professor who has time-traveled back from the future, and invaded the mini-beasts museum and transformed the creatures into large monsters. These monsters have become the professor's accomplices and are to aid him in his quest to take over the world. The monsters have been released, by the professor, into the wild and are causing disruption around the city and surrounding area. The game players must therefore use their 6PAQ "capture devices" in order to track

[1] Not related to Epic Games' Unreal Tournament

down and capture the monsters. Game players must collaborate as part of a team in order to successfully achieve this task.

4.2 Real Tournament User Interface

Each player's handset provides an active map on their mobile device that scrolls and rotates based upon the player's physical location and orientation. These are sensed dynamically from the GPS and electronic compass connected to the players handset (see below). Players have the ability to see their geographical surroundings, virtual monsters and team-mates position on their map.

Figure 1 – The Real Tournament user interface

4.3 Real Tournament Handsets

The Real Tournament handset known as the '6PAQ' (see Figure 3) is a customized Compaq iPAQ Pocket PC [7] with Compact Flash (CF) expansion jacket and a low power wireless 802.11b network card. In the design it was our intention to use a Trimble Lassen™ SQ GPS module, a digital compass and two micro-switches (used for firing and utility selection) and fit these into a gun which would provides an intuitive interface for the user and allows sensitive components such as the compass to kept level. All components are multiplexed via a PIC microcontroller and communicate via a standard RS232 serial interface to the iPAQ. The battery within the 6PAQ provides approximately 7 hours of continuous GPS and compass usage. While the iPAQ Pocket PCs and Socket wireless LAN cards enable approximately 3 hours of use.

The PIC microcontroller sends data using plaintext ASCII over the simplex 9600 baud RS232 connection to the IPAQ. Any contention for the serial line is resolved in order of real-timeliness. We define this order to be: trigger, electronic compass and GPS. GPS data is received and sent across the serial line once per second, orientation information is processed 10 times per second and the trigger status is received on change.

In addition to a novel interface for interaction the 6PAQ also provides a team audio communicator. This provides push-to-talk group communication functionality over the IPv6 wireless network – thus allowing players to synchronize their actions.

Figure 1.

(a) The 6PAQ gun cut-away view showing GPS, compass, trigger and battery

(b) A complete handset with iPAQ.[2]

Much of the Real Tournament code base consists of various components written using a combination of languages including Microsoft C#.NET and Embedded C/C++. C#.NET enables us to rapidly prototype client functionality and is used to maintain and control all game logic, whilst C/C++ facilitates the use of lower level functionality currently not supported by the .NET Compact Framework, such as serial port access, Mobile IPv6 communications and high speed graphics rendering. A third party graphics library, GAPIDraw [11], was utilized to provide graphics support for the game and FMOD [10] was utilized to provide sound.

[2] See http://www.mobileipv6.net/rt/handsets.htm for complete handset specifications.

4.4 The Aim of the Game

The basic aim of the game is to work with your team mates in order to capture as many monsters as possible during a game session. The highest ranked team is the overall tournament winner. To supplement the game play, specific regions of the park have been designated as challenge zones. Should a player enter such a zone, additional challenges may be carried out in order to increase the team's score.

We are currently publicizing the game to undergraduate and postgraduate students at the University and to local schools in the area. The aim is to trial the game by way of a series of "RT Rumbles" in which teams are pitched against each other over a series of months. This will enable us to achieve one of our main objectives, i.e., that of real user testing and evaluation, while at the same time providing the local community with access to next generation wireless technology, services and applications.

5 SYSTEM ARCHITECTURE

This section outlines the main decisions made while designing the key areas of the Real Tournament application; namely the wireless infrastructure and game architecture.

5.1 Wireless Infrastructure

Since the Mobile IPv6 Systems Research Lab is concerned with a wide range of issues relating to the development of mobile systems and applications, a major thrust of the work has been the deployment of a series of wireless 802.11b [15] networks around the University campus and city centre. Furthermore, the use of GPRS [19], as an "umbrella" for ubiquitous coverage across the whole area, has provided the project with a real-world testbed that enables experimentation with standard wireless overlay networks.

Figure 2 illustrates the logical network architecture of the Testbed network. The network infrastructure consists of the following entities:

- A *Wireless Overlay Network* around the city centre, the park and University campus based on different wireless network technologies including 802.11b and GPRS.
- *Mobile IPv6-enabled Client Devices* allow users to transparently roam between the wireless networks without losing connectivity or disrupting communication.
- *Access Routers* perform the layer-3 routing between the wireless and wired networks.
- The *Gateway Router* connects the Testbed networks via a high-speed link back to the campus backbone and the public Internet.
- An additional component developed as part of the Mobile IPv6 Research Lab is investigating access control in public wireless networks. The *Authentication Server* authorizes users with valid access credential to use the network. It provides the user's terminal with a valid access token, granting data traffic originating from that terminal access to the network [20].

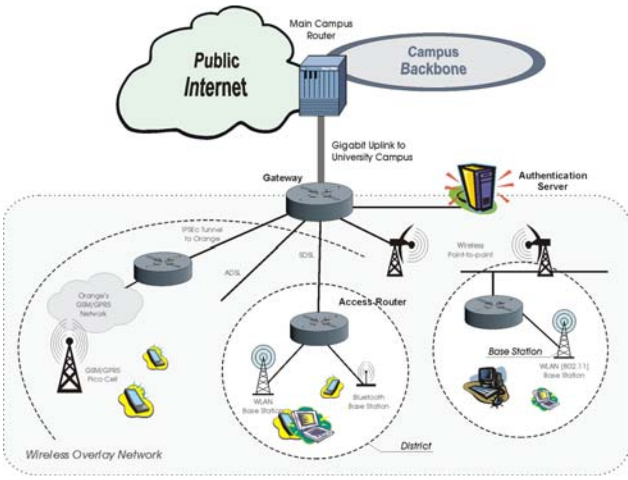

Figure 2 – wireless overlay network overview

Since the demonstration of the Mobile IPv6 protocol is of significant interest to our research, we designed the wireless network such that most of our 802.11b access points form different sub-networks in order to encourage handoffs [22]. Network layer routing between sub-networks is performed by the access routers. In order to keep the number of access routers to a minimum, whilst providing a large number of wireless (sub-)networks, we exploit Virtual LAN (VLAN) technology which is supported by most recent network switches and interface cards. VLAN allows a single router with a VLAN-capable interface to provide many virtual networks.

6 SOFTWARE ARCHITECTURE

There are a number of aspects which must be considered when designing a mobile real time application such as Real Tournament. Firstly, there is the question of how to provide a consistent view of the virtual world to all players. Secondly, the closely related issue of scalability must be addressed, so that many users may also interact without degrading overall performance. Finally, the mobile and distributed nature of the application means that the software architecture must be capable of adapting to changes in a clients network connection (e.g. from IEEE802.11 to GPRS).

In this section we justify the need for and propose a new gaming architecture capable of supporting large numbers of dynamic mobile nodes within a wireless overlay networked environment. First, we provide an overview of current architectures capable of supporting online, interactive gaming. Following this, we discuss an approach to scalable distribution of real time state along with consistency and synchronization methods.

6.1 A Survey of Game Interaction Topologies

A number of possible architectures exist which are suitable for supporting distributed multi-player gaming environments in fixed networks. We surveyed various groupware and online multiplayer gaming architectures in order to define a suitable model for sharing and communicating data between distributed sets of users. There are currently a number of well documented alternatives for CSCW [9] or gaming environments; in this section we focus on the centralized server, peer-to-peer and mirrored server [8] based architectures as potential candidates for wireless network operation.

A centralized server architecture makes use of a single server to store shared data and relay messages between clients. This approach supports high levels of consistency since a single copy of game state resides upon the server, which mediates user generated events, such as player and virtual entity location updates. An advantage of using this approach is that mobile clients send player generated events to the server, where they are processed, and transmitted to all relevant clients using area of interest filtering[8]. This has the effect of minimizing network utilization through aggregation and removal of redundant information. In the centralized server model, a user interface would only be updated once data is received from the game server. This not only ensures that consistency is strictly enforced, but also aids consistency between players since client disconnection can be determined quickly. The costs of this simplicity are the issues of including a centralized dependency. For example nodes that are topologically far away from the server have higher latency and therefore slower user interface feedback. In addition to this since state is stored on the server clients have no support for continued operation upon disconnection from the network which may be fairly frequent in a wireless setting.

A second alternative would be a replicated or peer-to-peer (P2P) architecture where each client maintains a replica of the game state. Here, there exists no central repository; instead, each client is responsible for maintaining a local copy based on messages disseminated during game play. Within this scenario, user generated events are updated by the client application locally before the event is distributed to all other clients. Upon receiving an event, remote clients update their local application data accordingly. The benefits of P2P architectures are two-fold. Firstly, they provide lower latency gaming since user input maybe handled locally rather than by processing through a central server. Secondly, since temporary disconnection in a wireless network is common it is useful to store state locally in order to continue operation. There are three main disadvantages to the P2P approach. Firstly, the increased complexity concerning issues such as consistency places a significant additional load on mobile clients. Secondly, bandwidth efficiency is reduced since there is no way to remove redundant data or aggregate messages through area of interest filtering as in a server approach. This is a significant problem in a wireless environment where network resources are more limited. Thirdly, P2P architectures do not scale well since the messages sent between peers increase exponentially according to the number of clients within the peer group. This further increases the inefficiency of network utilization adding further to the problems found within a wireless environment.

The mirrored server approach provides an interesting hybrid between the two previous approaches. With the mirrored server approach there exists multiple distributed servers. Each server is located topologically close to a collocated subset of the total number of players. Each game client communicates to the mirror closest to them in as in the client-server approach. The mirrors are then interconnected via a private low latency network (typically via multicast) reserved for gaming traffic effectively a P2P core. The advantage of this approach is that it shares the main advantages of both the client-server and peer-2-peer approaches as outlined above and in addition removes the single point of failure of the client-server approach. However once again there is little support for continued operation if a client is disconnected and mobile clients may still move away from a mirrored server thus negating the intended original advantage.

Upon evaluation of the techniques used in fixed networks we realized that in a mobile environment there is no guarantee that clients will be near a local server since mobile clients may move anywhere. This was one of the main flaws of client server related approaches. In addition, depending on the synchronization and consistency methods in use either a client will drift from consistency or if a more strict method is used all other nodes will be synchronized to the speed of the slowest peer. This behavior would be undesirable, particularly in a heterogeneous wireless network situation. However, minimizing bandwidth utilization was desirable to us if it could be implemented in some way.

On inspection of the peer-2-peer based approach scalability was almost certainly an issue since a fully connected P2P network is of N squared complexity. However the support for disconnection and minimized latency were desirable characteristics. Therefore we propose the development of a generic architecture which would support the following requirements:

- Low latency peer interaction.
- Permit the use of Mobile IPv6
- Scalable while maintaining performance and load balancing of network utilization.
- An approach that will supplement a high degree of consistency while maintaining a low latency characteristic.
- Support for disconnected operation through persistent state residing on multiple devices.

6.2 A Scalable State Distribution Architecture

One distinguishing factor that we must deal with is host mobility. The fact that the game clients are highly mobile means that the physical relationship between game players within the real world has a major effect upon the type of events they are interested in within the game arena. For example, players can only see artifacts, monsters and other players that are physically collocated. More specifically it is possible to create an area of interest that is shared between a subset of collocated players i.e. peer groups with area of interest filtering in peer-to-peer networks. Not only does this mean we can deal with scalability, it also means that bandwidth utilization is more efficient since peer groupings may be managed. Within these peer groups, game state must remain consistent, while events occurring outside of the immediate area are of lesser importance. A hierarchical approach enables state management to be simplified, thus reducing the amount of processing required by each mobile device.

More specifically, each mobile client communicates via the wireless network to a dynamically created peer grouping. To aid scalability, we are adopting a notion of ‘proximity’ in order to structure and scale the distributed state into a number of peer groups and maintain consistency at each level of the hierarchy, as shown in figure 4.

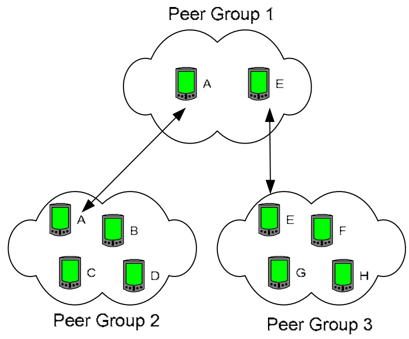

Figure 4 – A Hierarchy of Peer Groups

Associated with each peer group is a unit of proximity, the size of which will depend on the characteristics of the network technology being used and the requirements of the application. Factors which may contribute to this include performance, bandwidth, latency and location characteristics. For example. Applications may specify that clients with low bandwidth and a high latency be placed in small peer groupings while those devices with higher bandwidth and lower latency be placed in other larger groups. When considering Real Tournament, each peer grouping is created and maintained based on context obtained from the physical world, such as, player location and a players network connection eg wireless LAN or GPRS.

Within each peer group, there exists a super-node which has the added responsibility of forwarding highly consistent events which effect state outside a peer group to the next level up in the hierarchy. These might include events which affect the shape of the peer-group and may be forwarded up the hierarchy. One such example is a move event in which a player leaves a peer group to join another. Figure 4 summarizes the architecture and nodes A and E act as the super nodes for their respective peer groups. These super nodes are responsible for controlling membership to the group and for ensuring that specific events are relayed to other relevant peer groups.

6.3 Achieving Consistency and Synchronization

Within Real Tournament, the overall game play can be simplified so that the following generic events are sufficient to represent all changes in game state.

- Entity Move: A monster or a player changes location and/or bearing.
- Fire: A game player makes use of one of their utilities.
- Collision: A shot has hit a target
- Damage: A player or monster hit

In general, the mobile clients are generating both move and fire events and are required to calculate collision and damage events based on these. It is well known that these different events have different consistency and latency requirements [8]. Real time events have strict timeliness and lenient consistency requirements. Conversely, consistent events have strict consistency but lenient timeliness requirements.

Real time events must be distributed to other players immediately in order for entities to react adequately, for example, a move event. Delay in receiving movements will inhibit the effectiveness of player reflexes and perceived lag will have negative effects on the experience.

Unlike real-time events, consistent events affect the game state in a permanent and noticeable manner; therefore, consistent events must be consistently processed across all clients. For example, all clients must agree that a monster has been captured. Here, consistency is far more important than timeliness, so we might be willing to wait 500 ms before the monster disappears from the display.

To further aid the consistency within Real Tournament we have classified the state into three main categories. By prioritizing and processing the state information differently depending upon its class, we are able to control the consistency of real-time game state. The classifications are:

- **Producer Consistent State:** Producer consistent state is any information which can be assigned an owner and whose consistency is controlled by that owner. Thus, the producer of the state is regarded as the master and is therefore responsible for initiating events relating to changes to that state. For example, player location information can be regarded as producer consistent state since each player is generating regular events; therefore they can be regarded as the owners of the information. Ensuring other players also have accurate state information is therefore the responsibility of the producer.
- **Inconsistent State:** State or data items which are not controlled by any single entity but which exist in the game arena and be replicated by many peers are termed inconsistent. These normally have a lower priority within the game and are able to become inconsistent without affecting overall game play. For example, artifacts or power-up locations and collections, since they do not form a vital part of game play, are able to become inconsistent and do not need processing in real time.
- **Shared Consistent State:** Similar to inconsistent state, shared consistent items are not controlled by any single entity but exist within the game arena and may be replicated by many peers. These data items are considered a high priority for game play and therefore must remain consistent between all peers at all times. This state can therefore be regarded as jointly owned by multiple peers and every update to shared state must be agreed by all clients within the peer group. For example, monster movements within the game arena are not controlled by any single entity but are controlled using a voting/sampling method for determining conflicts. More specifically, each peer within a peer group must vote on the success or otherwise relating to highly consistent events such as monster hits, stuns or kill events. Each client posts an event to a well known multicast channel with is reserved for receiving and processing votes. Currently, a simple majority wins approach is being utilized by the system to determine the outcome of specific events. This method of maintaining consistency is one area we are currently in the process of developing further during our trial period.

To aid game consistency and prevent entities within the game from drifting away from the actual state of the game, we use timing information obtained via GPS to maintain synchronized clocks on each mobile device.

7 REAL TOURNAMENT GAMEPLAY

7.1 Monster Tracking

During normal game play, location and orientation updates are broadcast by each game client via application level multicast once a GPS fix has been received. Other player status updates are multicast upon change, such as a player pulling the trigger.

Of highest relevance to game play is the location of the virtual monsters as they roam the game arena and we have adopted an agent based approach to the management of monsters. In essence, each entity in the game (e.g. a monster) is represented by an object. During game play, monsters are dropped into the game arena and have a number of possible modes of operation, such as attack, wandering and flee. Currently, we have several relatively basic motion path algorithms for control which are used to determine a monsters current state, behavior and thus how they react to the environment.

During game play, the synchronized game clocks and control algorithms are used to move the monsters on a per second basis. Synchronization of monster movements is controlled by GPS updates, i.e. approximately once per second.

7.2 Capturing Monsters

Several different methods are available to players in order to successfully capture monsters during game play, these are, stun capture or kill. By detecting a trigger event from the 6PAQ device, a player's handset can determine when the player wishes to fire. By combining this event with a player's location, orientation and weapons charge (represented by a power bar on the game UI), we can determine the trajectory of a 'virtual laser' through the game arena. An extrapolation of this allows the system to determine whether the shot hits a target (e.g. monster) and spawns game events (such as damage). To determine a successful hit, the event is posted to the multicast channel and all peers within the group reply with their result. The current implementation enforces a majority win approach such that once a majority vote is reached no further replies are necessary. We intentionally created several monsters within Real Tournament which are largely immune to a single shot from a laser and may either require multiple hits before being captured or which require hits from different players to have any effect, i.e. one player must stun a monster and a second player is then able to capture it. This encourages players to act more collaboratively within the game and makes more use of the available collaboration tools, such as the audio communicator.

7.3 Group Communication and Collaboration

To facilitate cooperation between team players and further test the performance of the Mobile IPv6 stack with real-time traffic, we created an audio communicator. The current implementation operates around a "push-to-talk" model, whereby, each player must first press the voice record hardware button found on the Compaq iPAQ [7].

The audio communication system currently exchanges data using multicast channels which consist of a dynamically generated multicast address and port number pair. To facilitate communication between different groups, there is also a control channel. Only one member of each team is allowed to listen for control messages and is known as the super-node.

When initialized, each node sends a message to the control channel, containing the team name and team identifier. If a team channel already exists, the super-node will reply to the new node with the team channel details. If a team channel does not exist, then the new node assigns itself to be a super-node and creates a multicast address and port pair which forms the new team channel. The audio communication system makes use of a simple floor control protocol to orchestrate access to the media channel. A benefit of the guaranteed exclusivity of channel access is efficient network utilization since we ensure that only the client who has been granted the floor is allowed to transmit media and consume network bandwidth. The floor management task is assigned to each teams super-node. Regular control messages are distributed by the super-node and, should the case arise that the super-node fails, an election takes place among the remaining nodes and a new super-node is elected.

The audio stream consists of variable length packets, containing no more than 100 milliseconds of audio. The audio samples are currently digitized using Pulse Code Modulation (PCM) and consume 64Kbps of bandwidth. Although, this is a reasonable approach for use over the wireless 802.11b network, we will soon be making use of various low bit rate audio codecs so that we are able to make additional use of this service over the GPRS network.

8 INITIAL EVALUATION AND RESULTS

The Real Tournament application is still under active development and is now ready to undergo its first set of real user field trials. However, a series of preliminary user trials have been carried out during the development process in order to test the following elements of the game:

Control and Feedback: Providing adequate user feedback to the user of a mobile application is an area which we have experience with already [6], thus we were keen to evaluate whether or not users understood that the environment may fluctuate and that certain services (i.e. the audio communicator) may become unavailable at certain times during game play.

Physical and Virtual Navigation: Since the concept of capturing and mediating physical movements within the real world and mapping these onto a virtual world is an experience many have not tried, we were keen to evaluate how quickly this skill could be picked up and how responsive user's found the physical movement to virtual mapping to be.

Handset and User Interface: We were keen to ensure that, from an early stage, game players were able to use the handset easily and also ensure that the user interface was visible and usable within a mobile outdoor play environment.

To trial the system, we have adopted an iterative process by which each incarnation of the handset and client application has been tested by a selection of children from a local school close to the University and with post graduate students in campus. This process has involved demonstrating the hardware involved within the game before and after creating the handset to a small groups. More importantly, the use of navigation and player movement controlled by GPS was demonstrated, before showing that orientation is used in order to generate player direction. This provided intended users with a suitable mental model of the basic player controls.

The second set of trials were carried out at the Pervasive 2002 and MobiCom 2002 conferences during August and September 2002 respectively. Here we demonstrated a modified version of the game within an indoor environment. In this version, rather than utilizing GPS in order to move a character on screen, the Compaq IPAQ's one touch 5 way navigator button was overridden in order to move the character's position on screen.

The initial trials provided some very positive results. Firstly, all users of the system enjoyed the gaming experience and were very quick to adapt to physically moving within the real world in order to move their game character within the virtual game arena. Interestingly enough, both the children and academics that trialed the system almost immediately wanted the ability to 'shoot' other game players in addition to being able to 'capture' the virtual monsters, a feature we had not intended to include!

Largely, Real Tournament, the handsets and the underlying wireless network have performed well from early experiments. There are, however, several areas in which issues still exist. IEEE 802.11 coverage in the game arena is good, (approx. 95% by area), in areas of poor coverage we fall back to GPRS using Mobile IPv6. Currently performance issues are preventing us from utilizing GPRS effectively, as RTT of up to 4500 msec have been measured and average 700ms. Accuracy of the GPS is also in need of improvement (typically giving 4-5m accuracy) - we intend to develop differential GPS capable handsets in the near future. The digital magnetic compass proved to be very accurate, but susceptible to noise from its surroundings and pitch of operation.

9 RELATED WORK

Although there have been several different gaming architectures proposed and recent attempts at deploying mobile games, we are currently unaware of any system which aims to address the wide range of architectural considerations described in our work. Largely they focus upon prototyping the game concept itself, not upon how such applications would be developed, managed and provisioned over real wireless infrastructures. Specifically when considering wide area wireless IPv6 overlay networked environments for use by the general public.

Research within the field of virtual and augmented reality has recently begun to emerge into outdoor environments and aims to combine the virtual world of a game with the physical world in new and engaging ways. However, an approach does not yet exist for supporting mobile users within a wireless heterogeneous networked environment. This section provides a brief overview of the current systems available and outlines their respective research aims and objectives and how they differ to our work on Real Tournament and the MSRL Testbed.

9.1 Pirates!

The Pirates! project was a joint research collaboration between the Nokia Research Center (NRC) and the PLAY Institute [1]. Pirates! is a multi-player, indoor computer game for handheld devices connected in a wireless local area networked environment. In addition to network access, each handheld device is equipped with a custom-built sensor which is used to infer a players' relative proximity to other players and to locations in the real world. The players' movement between these locations triggers different game events and allows them to engage in a variety of activities such as exploring islands and taking part in sea battles [1]. The custom built location mechanism was used to explore the theme of proximity triggered interaction and how proximity based sensing could be used to provide richer game experiences in social settings. However, it was outside of the scope of Pirates! to consider its use in both an indoor and outdoor environment with a potentially large number of users. In this scenario, neither the custom built hardware infrastructure or software architecture would provide adequate performance given a large user base.

9.2 Can You See Me Now?

Can You See Me Now? [4] is a mobile mixed reality game in which up to twenty online players were chased across a city by three performers who were running through its streets. On-line players accessing a map over the Internet moved their characters across the city. Runners equipped with handheld wireless devices augmented with GPS receivers chased them by running through the city streets. Players were able to communicate with each other via text messaging. In addition, players could tune into audio streamed over the Internet from the runners' walkie-talkies. The focus of this work was an evaluation based on ethnography, discussions with game players and maintaining logs of all game activity which revealed design issues pertinent to other citywide mixed reality games [14]. While much can be learned from their experiences, our experiments are much more focused on systems level issues of deploying and utilizing a wide area wireless overlay network.

9.3 ARQuake

The ARQuake project [23], developed by the wearable lab at the University of South Australia, extends Quake into a system that allows users to play augmented reality games outdoors. Using a transparent head mounted display placed on a user's head, an internal half-silvered mirror combines images from an LCD display with the user's vision of the world [24]. By combining this display technology with a wearable computer, it is possible for the user to walk outdoors and visualize graphical objects that are not normally visible. This system currently requires each player to carry a very large amount of computer equipment and, furthermore, run a separate instance of the game arena. Communication or collaboration is not currently supported since the system does not include any network support. Our work focuses much more on providing services and applications to real citizens in the area and we therefore have to adopt technology users will be prepared to carry on a regular basis. Furthermore, a recent ARQuake user field trial revealed that the system had a number of limitations in terms of usability, including poor field of view, GPS tracking errors and disorienting 3D effects.

10 Conclusions and Future Work

This paper has introduced the work thus far as part of the industrially funded Mobile IPv6 Systems Research Lab. We have provided an overview of the wireless overlay networked hardware infrastructure and introduced the Real Tournament mobile game. The mobile game currently makes use of context such as location, orientation and player status in order to deliver a multimedia mobile gaming experience to local citizens in Lancaster.

We are currently in the process of attracting users to take part in a series of large scale and continuous trials throughout the spring and summer of 2003. These trials will allow us to refine both our hardware and software architectures further and also carry out network based performance trials of the Mobile IPv6 protocol, an area of significant interest to us. More specifically, the initial user trials will be used to thoroughly evaluate our hierarchical peer-2-peer model for consistency and support mobile game players. In addition, performance, latency and reliability results pertaining to the Mobile IPv6 stack will be of vital importance to our early trials. The results of these trials will then for the basis of future Real Tournament development.

We are currently in the process of building a new batch of Real Tournament handsets which include IPv6 communications over GPRS. To achieve this we are making use of the embedded Bluetooth device found within the Compaq iPAQ H3970 device which is communicating with a Sony Ericsson T39m cellular phone. Currently, we are using IPv6 tunnelling over the GPRS network although we are collaborating very closely with Orange in order to deploy native IPv6. The Orange IPv6 GPRS network will then form a key element within the game and enable game players to roam anywhere within the confines of the park and still be able to play the game, despite being disconnected from the wireless LAN. This wireless overlay will then afford a number of key research challenges pertaining to the 'best' or most efficient way of controlling and managing vertical handoffs.

REFERENCES

1. Björk, S., Falk, J., Hansson, R., & Ljungstrand, P., "Pirates! - Using the Physical World as a Game Board", Proc. Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction, July 9-13, Tokyo, Japan.
2. Brown, P.J., Bovey, J. D. and Chen, X., "Context-Aware Applications: From the Laboratory to the Marketplace", IEEE Personal Communications, 4(5): pages 58-64, October 1997.
3. Brown, P., J., "Triggering information by context", Personal Technologies, 2(1):1-9, September 1998
4. Can You See Me Now? Research Project, available via the Internet at http://www.canyouseemenow.co.uk/, 2001.
5. Cheverst K., Davies N., Mitchell K. & Friday A., "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project", Proc. MOBICOM'2000, Boston, ACM Press, August 2000, pp. 20-31.
6. Cheverst K., Davies N., Mitchell K., Friday A. and Efstratiou C., "Developing Context-Aware Electronic Tourist Guide: Some Issues and Experiences", Proc. CHI'2000, Netherlands, ACM Press, April 2000, pp. 17-24.
7. The Compaq iPAQ Handheld Devices Product Home Page. http://www.compaq.com/products/handhelds
8. Cronin, E., Filstrup, B., Kurc, A.R. and Jamin, S., "An Efficient Synchronization Mechanism for Mirrored Game Architectures", Proc. NetGames2002, April 2002.
9. Dourish, P. and Bellotti, V., "Awareness and Coordination in Shared Workspaces", In Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW '92), Toronto, pages 107-114. 1992.
10. The FMOD Music and Sounds Effects Project Home Page. www.fmod.org
11. The GapiDraw Project Home Page. http://www.gapidraw.com/
12. The GUIDE Project, Research Project, Lancaster University, available via the Internet at http:www.guide.lancs.ac.uk, March 2002.
13. Holmquist, L., E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M. and Gellersen, H., W., "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts", in Proc. of UBICOMP 2001, Atlanta, GA, USA, Sept. 2001.
14. Izadi, S., Fraser, M, Benford, S., Flintham, M., Greenhalgh, C., Rodden, T., "Citywide: supporting interactive digital experiences across physical space", in Dunlop and Brewster, eds, Personal and Ubiquitous Computing Journal, Volume 6, 2002.
15. LAN MAN Standards Committee of the IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications", IEEE Standard 802.11, 1999.
16. The LandMARC Project, Research Project, Lancaster University, available via the Internet at http://www.LandMARC.net, October 1999.
17. The Microsoft Mobile Developer Conference, MDC 2002, London Hilton Metropole, April 2002. Conference homepage available at http://www.microsoft.com/europe/mobdevcon/
18. Mobile IPv6 Testbed, Collaboration with Cisco, Microsoft and Orange, Lancaster University, available via the Internet at http://www.mobileipv6.net/, February 2001.
19. Nokia, "General Packet Radio Service - GPRS - Nokia's vision for a service platform supporting packet switched applications", White Paper, 1998.
20. Schmid,S., Finney, J., Wu, M., Friday, A., Scott, A., Shepherd, D., "An Access Control Architecture for Microcellular Wireless IPv6 Networks", in Proc. of 26th IEEE Conference on Local Computer Networks (LCN'2001), 14-16 November 2001, Tampa, Florida.
21. Shankar, B., Brown, A., MacMillan, D., ODonovan, P. and Wood, B., "Games and Entertainment Within the Mobile Phone Industry", 4 April 2002, available from www.gartner.com
22. Stemm, M., Katz, R.H., "Vertical Handoffs in Wireless Overlay Networks", ACM Mobile Networking, 1997.
23. Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W.,"ARQuake: An Outdoor/Indoor Augmented Reality First Person Application", In 4th Int. Symposium on Wearable Computers, pp 139-146, Atlanta, USA, Oct 2000.
24. Thomas, B. and Piekarski, W., "ARQuake: The Outdoor Augmented Reality Gaming System", Communications of the ACM,2002 Vol 45. No 1, pp 36-38
25. Weiser, M., "The Computer for the 21st Century", Scientific American, 265(3):94-104, September 1991
26. Sega, AT&T Wireless Push Mobile Gaming. Wireless News Factor Press release, available via the Internet at http://www.wirelessnewsfactor.com/perl/story/19673.html