ISPCell: An Interactive Image-Based Streaming Protocol for Wireless Cellular Networks*

Azzedine Boukerche, Richard Werner Nelem Pazzi, and Tingxue Huang

SITE - University of Ottawa, Canada
PARADISE Research Laboratory
{boukerch, rwerner, thuang}@site.uottawa.ca

Abstract. Remote interaction with immersive 3D environments with acceptable level of quality of experience has become a challenging and interesting research topic. Due to the high data volume required to provide a rich experience to the user, robust and efficient wireless transport protocols have yet to be developed. On the other hand, cellular network technology has been widely deployed and is growing fast. The provision of remote interactive 3D environments over wireless cellular networks has several interesting applications, and it imposes some unsolved issues. Node mobility creates unstable bandwidth, which is a problem when providing smooth interaction to users. Although PDAs and cell phones are low resource devices, which makes it prohibitive to load and render entire virtual environments, they can still render images with relative ease. Based on this idea, this paper proposes a streaming system which relies on an image-based rendering approach, and is composed of several modules: a packetization scheme for images, an image-based rendering approach based on view morphing and its corresponding RTP payload format, and finally a bandwidth feedback mechanism and rate control. This paper illustrates some of the problems faced in this area, and provides a first step towards their solutions. We discuss our algorithms and present a set of simulation experiments to evaluate the performance of the proposed schemes.

Key words: Interactive Streaming Protocol, Wireless Cellular Networks, Real-Time Protocol, Remote Virtual Environments, 3D Scenes, Low-Bandwidth Networks, Image-based Rendering, View Morphing

1 Introduction

For decades computer graphics technologies were trying to match virtual environments as closely as possible to real environments. The interaction with virtual environments has received a great deal of attention in recent years, which

* This work is partially sponsored by Grants from NSERC Canada Research Chair Program, Canada Foundation for Innovation and OIT/Distinguished Researcher Award.

includes video gaming, military simulations, medicine, architecture, product development and visualization, monitoring, virtual tours, and so on. The next step into the interactive virtual environment field was to enable users to remotely interact with the system, mainly to support collaborative interaction between systems and users. There has been a lot of interest and research on remote virtual environments issues lately, such as high quality graphical representations of the environments, system scalability, and low lag, just to mention a few. With the new development of wireless communication and mobile computing, an exciting research field has emerged, which enables users to interact with virtual environments using mobile devices, such as cell phones, PDAs, etc. Although, such thin devices have very low storage and processing power, thereby limiting the graphic rendering quality within these mobile devices. The dynamic nature of the bandwidth is another problem one has to face when providing remote interactive environments over wireless cellular networks. This issue, combined with the large amount of data required to be transmitted in order to load the environment on the thin device and later on to update the changes on the virtual environment, pose significant challenges in terms of network traffic, delay and storage.

In this paper, we propose an interactive streaming protocol (ISPCell) that enables users to interact with a remote virtual environment using their mobile devices. As opposed to the traditional geometry based rendering mechanism, this work focus on using an image-based rendering (IBR) technique on the client, which we refer to as view morphing [1], and requires low processing, as it renders novel views based on a collection of sample images. IBR algorithms are based on the plenoptic function [2]. View morphing is the simplest IBR algorithm as it relies on some geometric information about the scene, whereas lumigraph [3] and lightfield [4] use implicit or no geometry at all, which requires more processing.

IBR has some advantages over geometry-based rendering as it does not depend on the scene complexity, but on the image resolution, which fits perfectly to mobile devices. A client/server architecture is proposed and, basically, pictures are taken from the real world or from computer generated environments and stored on the server. The image acquisition is based on view morphing requirements, which needs some parameters such as depth information and position of the camera to render novel views. As the user navigates through the virtual environment, new views have to be displayed. The mobile device sends updates of the user position to the server. The server decides what images should be sent, based on the position of the user in the virtual environment. To improve the frame rate, a virtual user path prediction algorithm is proposed, allowing the server to pre-fetch some images to the client when enough bandwidth is available. The bandwidth feedback mechanism and rate control are designed to optimize the pre-fetching scheme, as its goal is to avoid starvation of images at the client side.

The interactive streaming protocol (ISPCell) is designed over the Real-Time Protocol (RTP) [5], which provides end-to-end delivery services for data with

real-time constraints, for instance, audio and video. According to the RTP specification, a new RTP payload format has to be defined for each different media type. Also, because the Real-Time Streaming Protocol (RTSP) [6] is intended for continuous media, a new streaming protocol had to be specified, which deals with interaction and random media access.

This paper is organized as follows: Section 2 gives an overview of the related work found on the literature. Section 3 presents the project architecture and its modules. The algorithms are described in Section 4. An extensive set of simulation experiments were performed and the results are shown and discussed in Section 5. Finally, conclusions and future work are provided in Section 6.

2 Related Work

This section reviews 3D rendering mechanisms that have been developed for mobile devices, followed by a discussion of related image-based rendering techniques.

2.1 Traditional Geometric Rendering on Thin Devices

Most of the work on 3D rendering on mobile devices has been done over the OpenGL ES API [7], and are limited to the mobile device hardware performance. Thus, the rendering quality is still poor, or in a very low level of detail. The Mobile 3D Graphics API (M3G), defined in Java Specification Request (JSR 184) [8], is another industry effort to create a standard 3D API for Java-enabled thin devices. A solution to visualize more complex 3D scenes on mobile devices is possible through Image-Based Rendering (IBR), which is discussed in the next section.

2.2 Image-Based Rendering

Mobile devices have lower processing power and memory than a personal computer. These constraints make it difficult to present complex 3D scenes on such thin devices. This is where IBR mechanisms take place. As IBR works with images, the rendering cost depends only on the display resolution, while the traditional geometry rendering cost depends on the polygon count of the scene. Therefore, IBR is appropriate for mobile devices.

The IBR methods are categorized based on the geometry information they require to render novel views. Some image-based rendering technics do not require geometric information. Lightfield [4] renders a new view by interpolating a set of samples. The problem with IBR methods that do not rely on geometric information is the huge storage and the acquisition mechanisms. Due to the huge amount of data required to be transmitted when rendering a large environment, a combination of image-based and geometry rendering should be employed.

This project is based on View Morphing [1], which is able to render any novel image by morphing two or more reference images. The basic principle is depicted in Figure 1. Images I_0 and I_1 are acquired at points C_0 and C_1 respectively with focal length f_0 and f_1. The novel image I_n, with focal length f_n, at point C_n, is rendered by the interpolation of images I_0 and I_1. There is also an image cache on the client in order to reuse a previously received image, significantly improving the system performance and reducing the network traffic.

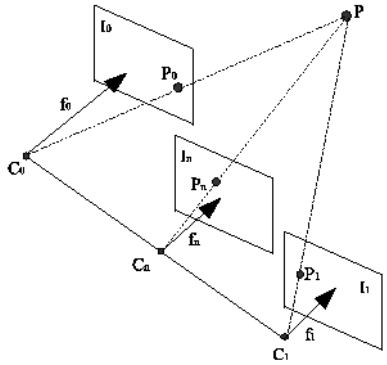


Fig. 1. View morphing with parallel views.

2.3 Remote Visualization of 3D Scenes

As discussed before, the visualization of complex 3D scenes on mobile devices can be improved by IBR . This subsection discusses some research works relevant to the solution proposed in this paper.

QuickTime VR [9] is the most popular image-based rendering system. But it is limited to panoramic scenarios, and the client device has to download the entire environment in order to start the navigation. ISPCell offers a higher level of freedom, as a user can walk through the environment, and there is no need to download the whole environment, as it is being constructed as the user visits different areas.

A client-server approach to image-based rendering on mobile terminals, similar to ISPCell, is presented in [10]. In this approach, the client send the user's position update messages to the server, which runs a 3D environment to take the necessary pictures and sends back to the client. The disadvantages of this work compared to ISPCell are that there is no pre-fetching mechanism, nor rate control, to make better use of the bandwidth, compromising the smooth interaction. Its main contribution in on the virtual camera placement algorithm.

Another work similar to ISPCell is discussed in [11]. This work relies on image-based rendering to enhance 3D graphics on mobile devices. The description of the client-server framework is very simple and, again, there is no mention of a pre-fetching, path prediction, or rate control mechanisms.

The next section introduces the project architecture and each of its modules.

3 The Proposed ISPCell Architecture

In this section, we shall discuss the main architecture of our ISPCell model, and its main components, which include: a modified JPEG codestream, the new RTP payload format for view morphing, the ISPCell streaming protocol, the bandwidth feedback mechanism, the rate control scheme, and the path prediction and pre-fetching algorithms.

3.1 Modified JPEG Codestream

A new JPEG codestream scheme was specified due to the error-proneness wireless channels. Figure 2 shows the new JPEG codestream. A packetization item is an atomic component of the new JPEG codestream, for instance a main header, a layer header, or a pixel block. Before being sent to the network layer, the image codestream is split into packetization items and encapsulated in RTP packets. The main issue behind this packetization scheme is to avoid that errors in one packet affect other packets, by keeping the packetization items independent from each other.

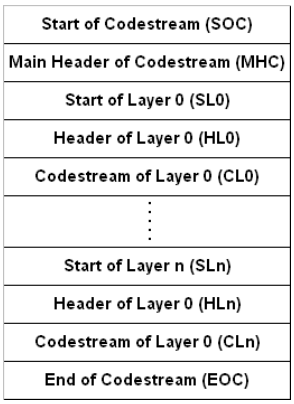
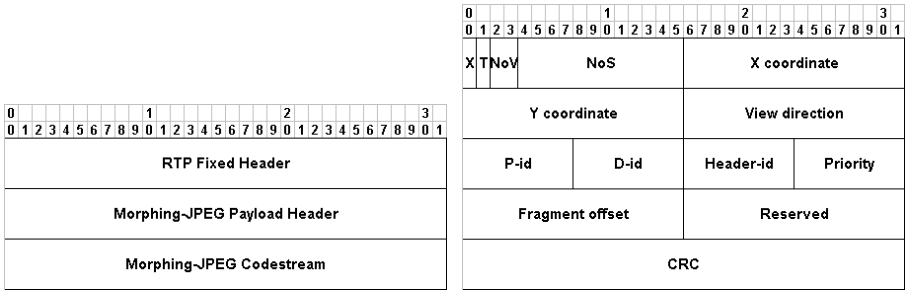


Fig. 2. Structure of the new JPEG codestream

3.2 A New RTP Payload Format

The RTP protocol carries payloads of real-time applications, such as interactive navigation through virtual environments. For the streaming of any multimedia compressed data over wired or wireless networks, new payload formats over RTP are required, such as H.26x over RTP [12, 13] and G.7xx over RTP [14, 15]. Therefore, a new payload format for view morphing over RTP was specified, as



(a) (b)

Fig. 3. (a) Structure of a RTP packet for View Morphing. (b) View Morphing payload header.

well as a new packet header for view morphing, as depicted in Figure 3(a) and Figure 3(b) respectively.

The RTP fixed header fields are the same as those of other RTP application payloads. Only the following fields are exceptions, and are specific to the View Morphing algorithm:

- **Payload Type**: according to the RTP standard, this field specifies the format of the RTP payload and determines its interpretation by the application [14].

- **Number of Video Unit**: the transmission stream is constructed from a series of video units consisting of one or more images. In general, a video unit is divided into more than one fragment. Number of Video Unit (NVU) is a random value given to all RTP packets of a video unit. This field helps the receiver identify the complete transmission of a video unit.

- **Timestamp**: The View Morphing stream has no strict sampling instance. Unlike other media types, the timestamp for View Morphing does not indicate the sampling instance. Nevertheless, it is significant for calculating synchronization and jitter when other media streams are associated with View Morphing.

The View Morphing payload header fields are described bellow:

- Payload header extension **(X)**: this bit flag is activated when there are some supplementary information following the payload header.

- Twin images **(T)**: this bit is set to inform the renderer to process two images, otherwise the rendering algorithm can render the image directly.

- Number of images in a video unit **(NoV)**: informs whether the packet contains zero, one or two images.

- Number of an image in the streaming **(NoS)**: works as a sequence number to make sure all image fragments were received at the client.

- Viewpoint **(X,Y)**: coordinates of the viewpoint.

- View direction **(VD)**: angle that indicates the view direction.

- Viewpoint and view direction identification **(P-id and D-id)**: these fields help identify the viewpoint and view direction when data is corrupted or lost.

- Codestream header identification **(header-id)**: helps recovering the main header when data is corrupted.

- Priority of image layers **(Priority)**: Morphing-JPEG supports layered compression. The server can send quality layers separately, so the lower the layer, the higher the priority.

- Fragment **offset**: to reassemble the codestream.

- **Reserved**: field reserved for future use.

- **CRC**: detects whether the payload header is corrupted or not, which part is corrupted and tries to correct some bits.

3.3 The Interactive Streaming Protocol Algorithm

Basically, the construction of an RTP packet begins with the assignment of the payload type field for the Morphing-JPEG. If the current packet is the first fragment of a video unit or it is the whole video unit, then check if this is the first packet for that session. If it is the first packet, the field NVU and timestamp in the RTP header receive a random value, that will be the first value of a sequence number. If this is not the first packet, NVU is incremented by one. All other fields are set according to the specification. For instance, the field X is set to 1 if there is an optional payload following the payload header; the field NoV is set according to the images the client will have to process, for instance, it will be set to 00 to instruct the client to process the first image on that video unit, 10 for the second, and 11 for the last one.

The algorithm for the client is just a parser for the server codestream. It first checks if the payload type field is set to Morphing-JPEG, and then checks the NVU field to see if it is different from the last one. If it is different, instructs the application layer to render the image at that moment. Then the algorithm gets the timestamp to calculate the synchronization and jitter. If the field X is set to 1, it will locate the optional payload header, parse the codestream and send it to the application. For the CRC scheme, the algorithm verifies if the viewpoint and direction are correct. If they are correct, the algorithm proceeds checking the priority and the remaining fields. If not, the algorithm gets the approximate values for viewpoint and direction from P-id and D-id, and runs the CRC on them. The final step is to check the offset field. If the offset is equal to the last offset plus the length of the packetization data, then it just appends the packetization data to the previous one. If that is not the case, the algorithm waits for a short timeout period. If the delayed packet does not arrive, it informs the server that an RTP packet was lost, which will adjust the parameters related to the feedback mechanism according to the reports of RTCP.

3.4 A Bandwidth Feedback Mechanism

The conventional RTP provides periodic feedback on the quality of data distribution through the RTP Control Protocol (RTCP). The RTCP is intended

for wired network infrastructures, in which the available bandwidth changes smoothly. Therefore, the periodic transmission of control packets is enough to control the adaptive encodings and the speed of data distribution. However, wireless network bandwidth changes dramatically. The periodic feedback mechanism can not accurately and promptly reflect the situation of a wireless channel. The immediate feedback mechanism described in this paper involves sending ACKs for every received RTP packet. By keeping track of these ACKs, the server can establish the network status. The ACKs are kept as small as possible with only its type, a sequence number for ordering, and a timestamp. The timestamp is used to approximate the round trip time (RTT) of each RTP packet. Missing acknowledgments are interpreted as dropped RTP packets. Following is the description of the algorithm.

When the server receives an ACK, it will parse the RTP packet and extract the sequence number and timestamp. Then, the server adds the sequence number to a list of successfully transmitted packets. The server then calculates the RTT and adds it to the list of recently transmitted packets. For each recently transmitted packet, if the sequence number does not exist in the list of successfully transmitted packets, and if the RTT of the previous packet minus the RTT of the next packet is greater than an acceptable variation value, then the number of packets lost is incremented by one. If the number of packets lost is greater than 0, then the network status is set to congested. Otherwise, if the RTT of the last received packet minus the RTT of the first received packet is greater than a threshold, then the status is congested. If the RTT of the first received packet minus the RTT of the last received packet is greater than a threshold, then the status is unloaded. Otherwise the network status is constant.

3.5 The Rate Control Algorithm

The rate control mechanism is used by the server to make a better use of the available bandwidth. It is based on the reports from the feedback mechanism. If the network status is congested, the rate is decreased. If the status is unloaded, the rate is increased, and if the status is constant, the rate is left unchanged. The increment value is crucial for the performance of ISPCell. It only makes sense if it can stream an extra image, and not only an extra packet or so. Upon receiving a request, the server will determine the network status. If the network is unloaded, the rate is increased by an amount corresponding to one image. If the network is congested, the rate is decreased by an amount corresponding to one image.

3.6 Virtual User Path Prediction and Pre-Fetching Schemes

The path prediction mechanism is based on previous and recent virtual user's movement in the environment. There are two navigation modes: linear or rotational. For instance, if the user is moving along a straight line, and based on previous movements, the path prediction mechanism will probably infer that

the user will continue on the same path. In the case of rotation in the virtual environment, the path prediction will get the nearby positions based on a threshold angle.

Based on the path prediction mechanism, the server will pre-fetch some images to the client device. The pre-fetching algorithm takes advantage of available bandwidth to send images in advance to the client, saving some requests and network traffic, improving the frame rate on the client.

4 Simulation Experiments

ISPCell was implemented and simulated on the NS-2 [16] network simulator. The cellular network scenarios consisted of several base stations, a streaming server, and mobile nodes ranging from 10 to 100, moving at 5m/s to 20m/s, in an area of 1500x1500 m^2. An image set composed of a thousand images was built on the server. Also, a request generator was implemented on the client, which requests those images from the server. In order to evaluate the performance of the proposed interactive streaming system, the following metrics were used: packet loss length, packet loss duration, packet burst length, packet burst duration, packet rate and packet delay. To evaluate our ISPCell path prediction scheme, we have chosen the following performance metrics: average number of images in the cache versus the number of requests.

4.1 Results and Analysis

Examining the results from Figure 4, one can notice the adaptability of the rate control mechanism reacting to the reports from the feedback mechanism. When one client was connected to the server, the amount of images in the clients cache is to 309 for only 150 requests. The control mechanism was aware of the available bandwidth, and was quick to utilize it by sending additional pre-fetched images. If the pre-fetching was not implemented, the number of images at the clients would not go over 150 (ratio of 1 image per request). Having such a large number of images for a little number of request helps making the streaming run more smoothly. When considering fifteen clients connected to the server, the bandwidth dropped. Again, the rate control mechanism was able to adjust to the new scenario and reduce the streaming rate. For the same number of requests (150), the server only managed to stream 176 images; almost half when the compared to one connected client.

Packet loss length indicates the average number of lost packets. The packet loss duration represents a period during which the average number of packets are lost. With a higher traffic, more packets are lost. However, depending on how the packetization item is grouped and if an appropriate packet length is applied, the packet loss length will be reduced. The packet burst length and the packet burst duration represent the burst performance of the RTP. A burst is a period during which a high rate of packets is lost.

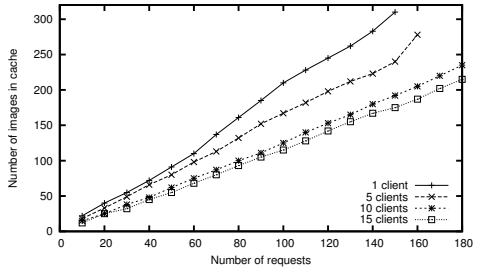


Fig. 4. Performance of the pre-fetching mechanism.

The number of handoff increases along with the traffic load. Therefore, *packet loss length* and *packet loss duration* also increases, as can be seen in Fig. 5(a) and Fig. 5(b) respectively. It can also be noticed that the speed of mobile hosts does not heavily affect the *loss length* and *the loss duration.*

The *packet burst length* and *packet burst duration* represent the streaming feature of an error-prone communication channel. During the burst period, RTP packets can not reach the destination. The burst is mainly caused by the handoff. Fig. 5(c) and Fig. 5(d) show the burst performance of the ISPCell. These two metrics do not depend on the node speed but on the node density. Both *packet burst length* and *packet burst duration* increase based on the fact that hand-offs are successfully completed with more difficulty, as the number of nodes increases.

The *packet rate* metric depends on several factors including the sender's transmission speed, the network communication bandwidth and the packetization scheme. As depicted in Fig. 5(e), the packet rate decreases as the number of nodes increases, due to the extra traffic.

As shown in Figure 5(f), the packet delay stays around a stable value of 110 milliseconds and is not affected by the traffic and the node speed, which is good for the kind of application presented in this paper. Interaction requires fast protocols to maintain a smooth frame rate for a reasonable user quality of experience.

5 Conclusions and Future Work

Remote interaction in high quality virtual environments on mobile devices have promising applications. However, due to the large amount of 3D data to be transmitted, the drastic changes in bandwidth, as well as the error-prone wireless channel, streaming 3D data through wireless cellular networks represents one of the most challenging problem that one has to deal with. In this paper, we have proposed a new RTP protocol for view morphing rendering over wireless last-hop scenarios, which we refer to as ISPCell, and that consists of the new RTP header structure, the packetization scheme, and the immediate feedback mechanism.

We have described our proposed ISPCell protocol, and we have presented a set of simulation experiments to evaluate its performance using ns-2. Our results indicate that ISPCell exhibits a good performance using several realistic scenarios. In the future, we plan to investigate further the performance of our scheme using a probabilistic virtual path prediction, which we believe has the potential to improve the system's performance.

References

1. S. M. Seitz and C. M. Dyer. View morphing. In Computer Graphics Proceedings, Annual Conference Series, pages 2130, Proc. SIGGRAPH96 (New Orleans), August 1996. ACM SIGGRAPH.
2. E. H. Adelson and J. Bergen. The plenoptic function and the elements of early vision. In Computational Models of Visual Processing, pages 320. MIT Press, Cambridge, MA, 1991.
3. S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In Computer Graphics Proceedings, Annual Conference Series, pages 4354, Proc. SIGGRAPH96 (New Orleans), August 1996. ACM SIGGRAPH.
4. M. Levoy and P. Hanrahan. Light field rendering. In Computer Graphics Proceedings, Annual Conference Series, pages 3142, Proc. SIGGRAPH96 (New Orleans), August 1996. ACM SIGGRAPH.
5. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. Standards Track, Network Working Group, January 1996.
6. H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and A. Narasimhan. Real time streaming protocol (rtsp). Inernet Draft, Internet Engineering Task Force, February 2004.
7. OpenGL Embeded System. http://www.khronos.org/opengles/ 2006
8. Java Specification Request 184 (2005) - Mobile 3D Graphics API for J2ME http://www.jcp.org/en/jsr/detail?id=184
9. S. E. Chen. QuickTimeVR an image-based approach to virtual environment navigation. Computer Graphics (SIGGRAPH95), pages 2938, August 1995.
10. G. Thomas, G. Point, and K. Bouatouch. A client-server approach to image-based rendering on mobile terminals. Technical Report, ISSN 0249-6399, France, January 2005.
11. CHANG C., GER S.: Enhancing 3d graphics on mobile devices by image-based rendering. In Proc. 3rd IEEE Pacific-Rim Conference on Multimedia(2002).
12. T. Turletti and C. Huitema. Rfc2032: Rtp payload format for h.261 video streams. Stardards Track, Network Working Group, Octorber 1996.
13. C. Zhu. Rfc2190: Rtp payload format for h.263 video streams. Standards Track, Network Working Group, September 1997.
14. H. Schulzrinne and S. Petrack. Rfc2833: Rtp payload for dtmf digits, telephony tones and telephony signals. Standards Track, Network Working Group, May 2000.
15. R. Zopf. Rfc3389: Real-time transport protocol (rtp) payload for comfort noise (cn). Standards Track, Network Working Group, September 2002.
16. The Network Simulator. http://www.isi.edu/nsnam/ns/

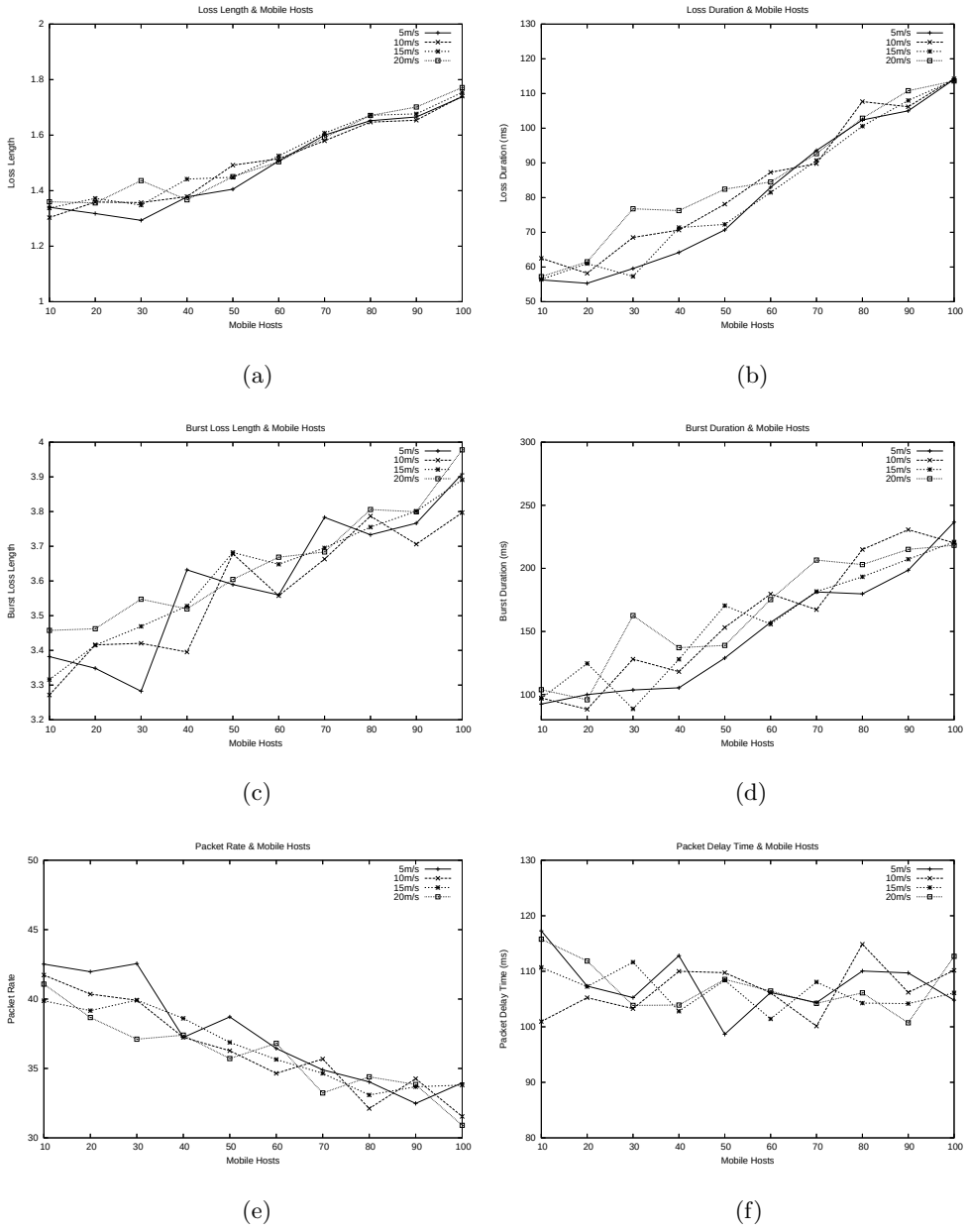


Fig. 5. (a) Packet loss length. (b) Packet loss duration. (c) Burst length. (d) Burst duration. (e) Packet rate. (f) Packet delay.