Perceptual Network Metaphors: Involving the User in the End-to-End Argument

Manuel Oliveira **Mel Slater**
Department of Computer Science
University College London
Gower Street
W1CE 6BT, London
{m.oliveira, mel}@cs.ucl.ac.uk

Jon Crowcroft
Computer Laboratory
Cambridge University
jon@cl.cam.ac.uk

ABSTRACT

The trend in existing networked applications has to been to isolate the application from the network. However, such an approach antagonizes the users of virtual environments and online games who acknowledge the existence of the network and attribute to it any disruptions to their sense of enjoyment. This paper presents a novel approach to the impact of degraded network performance on people's experience of virtual environments and online games. Contrary to the customary approach, rather than hiding the state of the network, the application monitors its state to determine a set of global and local properties that are used to manipulate visual aspects of the displayed scene - Perceptual Network Metaphors (PNM).

With PNMs representing the current state of the network, the idea is that it may be possible to influence people's expectations and behavior, in similar way that in everyday life people naturally change their behavior according to the weather. This approach brings the user into the end-to-end argument, thus complementing the existing network compensation techniques.

The paper describes the PNM conceptual framework and the operational architecture of the associated PNM Manager.

Keywords

Application level framing, end-to-end argument, virtual environments, metaphors, online games.

1 INTRODUCTION

The World Wide Web (WWW or Web) continues to be the prominent source for generating network traffic [Arlitt] on the Internet. The communication architecture is based on the client/server model, where the clients provide users with a window for visualizing the data residing on multiple remote servers geographically distributed. The entire repository of data is structured as hypertext, where a document has embedded within various elements. The client application is required to request the elements from their remote sources and integrating them together on the local host, ultimately presenting to the user with a webpage. The request/response protocol requires TCP as its transport protocol. Whilst the Addictive Increase Multiple Decrease (AIMD) mechanisms of the protocol are largely responsible for the relative stability of the Internet, despite its growth, the application is successfully isolated from the network without any understanding of its state at any particular instance. Consequently, the application is oblivious to the problems that may afflict the response time associated in the retrieval of data, which in turn leaves the user wondering what may have occurred to stall the application. Current browsing client applications provide some means of conveying the user some primitive visual feedback in the form of an hourglass or time progress bar. Both approaches do not operate upon the actual state of the network but rather on the amount of bytes of particular resource remains to be retrieved.

The primitive visual feedback mechanism suffices for the WWW, where the interaction paradigm of browse/request/retrieve closely resembles the underlying communication infrastructure. However, a new genre of networked applications, namely online games and virtual environments, has emerged and experience increased popularity. These applications have more complex interaction paradigms coupled with the fundamental requirement of real-time interactivity. In this case, the users are less accommodating to problems experienced on the network and the impact on their experience whilst interacting with the application. To support this claim a small online survey [url] was carried out aiming for the user population that played First Player Shooters (FPS) such as Half Life [], and Massive Online Role Playing Games (MORPG) such as Everquest []. The online survey was designed with 23 questions addressing issues of feeling immersed in an alternate reality (Presence [Biocca]), sharing the virtual space with other people (Co-Presence [Schroeder]) and the perception regarding the network impact upon the application. The results[1] demonstrate that the users from the gaming community, unlike with web browsing, have acquired some fundamental understanding of the network and attribute most of the disruptions in their enjoyment to some problem that may be afflicting the network at the time. Evidently, the network mechanisms to

---

[1] The questionnaire used for the online survey and a table summarizing the 193 responses may be encountered in Appendix

accommodate and smooth the impact of network problems are not entirely successful and a new paradigm is required to avoid the disruptions experienced by users.

This paper presents a new paradigm addressing the aforementioned shortcomings of existing real-time networked applications, namely online games and virtual environments. Analogous to the case of network modeling, where the traditional approaches have been erroneous due to the traffic being self-similar [Paxson], it is necessary to involve the end-user in the model [Crovella, Feldman]. Our approach consists of employing metaphors to circumvent many of the problems caused by network problems in either virtual environments or online games, namely delay. The fundamental idea of this work is that information, embedded as a natural part of the virtual environment, is made visible to users, indicating the local and global state of network performance.

In everyday life people naturally modify their behavior in response to environmental factors such as the weather. When it is raining people do not play outdoor tennis, and do carry umbrellas. When it is foggy people drive more carefully and attempt to signal their own position to others in an attempt to improve safety. Likewise, poor network conditions can be indicated by, for example, changing the virtual ‘weather’ in the virtual environment. Rain in cyberspace, for example, may mean that network conditions are bad. User expectations and behavior are naturally modified, without necessarily impacting the overall sense of presence within the environment.

The remainder of the paper is organized into six other sections. The next section will briefly describe the disparity between the spatiality of a virtual environment and the underlying network infrastructure. When presenting the network QoS cyclic model, some of the most popular network techniques used by both virtual environment and online games will be presented. The Perceptual Network Metaphors section presents the core concepts of the novel approach, supported by some descriptive examples. The following section describes the main blocks of the conceptual framework of the metaphor functionality. This is followed by an overview of the actual architecture of the implementation. A brief description of a prototype implementation is given in next section. Finally, some concluding remarks are given along with necessary future work to validate the effectiveness of particular metaphors with end-users.

2 VIRTUAL ENVIRONMENTS

In virtual environments and online games, the world along with all its entities is contained within a database that is shared amongst all the participants. To create the illusion of shared virtual space, this database must be perceptually consistent across all the users. Therefore, any update to the database, such as a user moving, must be disseminated to all the users.

One of the focal requirements of a virtual environment is to provide and maintain the illusion of real-time interactivity, without any disruptions occurring. The fulfillment of this requirement is constrained by how fast information travels along the underlying network supporting a virtual environment. Unfortunately, the physical distance of the network route does not correlate to any spatial relationship within the virtual environment, which is what represents the perceived reality. This may result in serious problems that ultimately disrupt the sense of presence [4] and co-presence [25] of a person.

When considering social interaction between avatars in a virtual environment, it is expected that the information dissemination is nearly instantaneous because of the small distance between the stimulus and receiver. However, as pointed out in [9], this is not the case when the clients are located on exact opposite earth locations (22417.5km): the events thus generated will have a latency of 100ms, albeit the propagation speed ($2x10^5$km/s) being close to the speed of light ($3x10^5$ km/s). Evidently, this is an optimistic scenario of a single optical fiber connecting both remote sites, but in reality there are a series of interconnecting sub-networks that add to the overall delay. This amount of latency may become perceivable by the users, if a particular threshold is exceeded [1].

The impact that the network may have on the experience of the users while immersed in the virtual environment has been widely recognized [7, 17, 18, 21, 30]. Within the Internet Engineering Task Force (IETF), there has been an earnest effort [6] to improve the current best effort given by the Internet by extending the current service model to provide Guaranteed Service [27] and Controlled Load [34].

Should the virtual environment be of reduced dimensions with few participants, analogous to the scenario of a videoconference, it is possible to benefit from the allocation of differentiated services [Yu]. However, with large scale virtual environments involving multiple participants, the mechanisms involved to reserve resources on the network, such Resource reSerVation Protocol (RSVP) [33], are not suitable for the volatile nature of application due to the overhead of resource negotiation.

3 NETWORK QOS CYCLIC MODEL

The current trend in virtual environment has been to compensate and shield the user from the state of the network. At any given instance, the application determines the Quality of Service (QoS) that is available from the network and may adapt if any mechanism exists. This ongoing negotiation process is known as the Network QoS Cyclic Model and is illustrated in Fig. 1. The user at any instance is left unaware of the state of the network.

Fig. 1 - Standard network QoS cyclic model

In [8], network resources are reserved in advance to accommodate the QoS requirements of the virtual environment system. This is done statically at the beginning of the session, which implies that resource management is non-realistic for virtual environment due to its ever-changing nature.

Due to the overhead of the integrated service model [6], most of the virtual environment systems and online games have adopted the approach of using network compensation techniques, of which some of the most popular of which are summarized in the following subsections.

3.1 Dead Reckoning

With the exception of audio/video streams, the movement updates of a user, while navigating around the virtual environment, probably constitute the majority of the data traffic generated by the application. Considering that the updates are ever changing, the focus is on interactivity rather than causality. So the approach is to have state regeneration where state updates are generated at a given sample rate. The problem of an update loss by the network is rectified by the fact that the following update renders it obsolete.

The problem with state regeneration is that the traffic generated may be considerable, as the initial version of the Doom™ [22] game demonstrated. The adopted solution was to disseminate one packet per keystroke, which quickly inundates the local network. The solution to reducing the sample rate is to use dead reckoning, where every client predicts via a simulation model the next position of the remote clients. The simulation is synchronized with the actual position by receiving a sample or a correction update.

The problem with dead reckoning is determining the balance between sample rate and traffic generated. If the sample rate is low, then inconsistencies may occur [19]. The impact of these inconsistencies may be countered by resorting to convergence techniques [28], but these have a perceptual threshold, beyond which it becomes evident to the user that something is wrong.

Dead reckoning addresses the problem of packet loss and throughput.

3.2 Client Prediction

There exists a wide range of infrastructure architectures to support a virtual environment. Each approach has its advantages and disadvantages. The core idea is that all the processing of the world is done at the server based upon the updates received from clients. In turn, the server then communicates to all the clients their state. Although client/server architectures enforce consistency due to the central nature of the database, they also introduce additional latency that may be detrimental for real-time interactivity.

Current online games have countered the problem by delegating some of the processing to the client by means of Client Prediction [2] techniques. These methods are based on the assumption that the client may proceed with an operation because the server will most likely validate it. If this is not the case then the server will inform the client and it will have to do a rollback.

In distributed architectures, client prediction may be used to determine lock ownership transfer [35]. The core objective of client prediction is to counter the impact of latency.

3.3 Buffering

The objective of a virtual environment is to convey the illusion of sharing the same space amongst multiple participants. This implies that events should seem to occur at the same time across all the clients involved. Considering the speed propagation of the network and the disparity with spatial proximity, obtaining instantaneous dissemination is not feasible.

A technique [12] that ameliorates the effects of latency is based on a time buffer processing mechanism. Each element of the buffer corresponds to an interval of time when all events are processed. Naturally, this implies that all clients are synchronized according to the same clock.

The essence of the approach is to avoid immediately processing the local events and to add an artificial delay similar to the latency that remote clients will experience when receiving the event. This is feasible so long as the artificial delay introduced does not reduce the perceived responsiveness of the user interface.

In [21], it was shown that users could adapt to latency provided that it remained consistent. With buffering it is possible to reduce the effects of latency variance.

Buffering addresses latency and its variance on the network.

3.4 Time Distortion

In a virtual environment, each user has the illusion that their reference for time is absolute, with everything else happening in accordance to it. Being able to maintain this impression will reinforce the user's sense of presence.

There exist several techniques that exploit the notion of time by either expansion or contraction as deemed necessary. A simple example may be found in the 2D ping-pong game involving two users confined to a rectangle field. The objective is to score a point by dispatching a ball beyond the opponent's defences. Each user controls a paddle and tries to hit the ball back across the field towards the opponent. Although the trajectory of the ball is deterministic according to physics, determining its exact position in time taking into account two different reference points is non-trivial. This is due to the existence of network latency. A possible solution [16] is to render the ball in real-time according to the user that will interact with it while simulating it with a certain delay for the other user. The roles reverse once a user hits the ball back across the field.

The ping-pong scenario works due to the fact that only two users are involved and constrained in their movements to a well-defined path. In a virtual environment, it is not feasible to constrain the user in similar fashion, thus the need for generalization [24, 26] based on relativity. However, current research has focused on small sized user groups with a controlled network environment.

Another way of distorting time is by making consistency roll backs [31] in the virtual environment database to a well-known synchronization point whenever inconsistencies arise due to latency problems. However, this approach results in disconcerting experiences to the users as the Half-Life™ [15] game clearly demonstrates by the coined term of "shooting around the corner" [2]. This phenomenon typically occurs when clients have disparate latency times. The usual example is when a user scores a hit on another one whose client experiences much lower network latency. This results in the victim having the impression of being shot even though they were already around the corner. This situation occurs because the firing event is received with sufficient delay for the target user to have moved elsewhere.

With time distortion it is possible to address the impact latency may have in the consistent perspective between different participants.

4 PERCEPTUAL NETWORK METAPHORS

While network compensation techniques will always be a necessary and useful part of a system, every technique has operational thresholds beyond which the detrimental effects of the network have an impact on the user.

These problems arise because the user has been shielded from the effects of the network. The break in their cognitive perceptual model occurs because something unexpected occurs that cannot be easily explained with the information available. The problem is aggravated in a distributed application where the users share a common experience, because the incident is not isolated to a single user and ripples across the other users.

In of particular resource, it is reported that end-users have some notions regarding the network. The study carried out tried to assess the users' perception of QoS when interacting with commercial websites. Although the target application was online shopping, the results may be applicable to other application domains, including virtual environments. The results demonstrate a more favorable response from users when a better understanding of the network state is conveyed explicitly in some form.

In virtual environment, a preliminary study [30] was carried out to evaluate the behavior of users when playing a simple 2- player ball game in the presence of various degrees of network delay. The result of the study demonstrated that users not only perceive the network delay beyond a given threshold, but also develop and adopt behaviors aimed at compensating the problems encountered. These results were used to implement a visual widget that consisted of a wire-framed volume that encompassed an avatar experiencing network delays [14]. The assessment of the current delay was visually displayed as a sliding bar on top of the volume.

A more focused study [11] was carried out to investigate the use of visual cues to provide information regarding the impact of network latency. The virtual environment would explicitly convey the user with state of lock ownership acquisition. The approach consisted of allowing people to manipulate replicas of the original object until ownership resolution was achieved. Once the owner was known, the respective object would become solid and all other instances would fade. So a user would be allowed maximum responsiveness, but with the knowledge that their manipulation may not affect the actual object.

In all of the above [5, 11, 14, 30], the aim is to involve the user in the enhanced Network QoS Cyclic Model, as illustrated in Fig. 2.

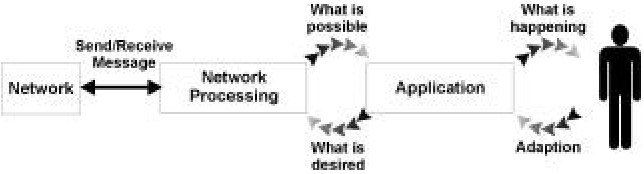


Fig. 2 - Enhanced network QoS cyclic model

This approach provides the user with further information that allows them to extend their conceptual perception of the virtual environment. Therefore, when serious problems occur in the network and the network compensation techniques fail, the user will be able to adapt.

Considering the importance of sense of presence and associated co-presence within a shared virtual environment, it is fundamental that the presentation of relevant information be done in the least intrusive fashion possible. Thus this paper proposes the use of Perceptual Network Metaphors (PNM)s, which are seamlessly integrated into the context of the virtual environment such that the user does not distinguish that the underlying system is conveying network data.

Fig. 3 illustrates the taxonomy of the various actors, roles and relationships within a virtual environment using PNM. These notions will be used throughout the remainder of the paper.

Each PNM embodies the relationship between a host and the underlying network. However, from the perspective of a particular client there are two main types of perceived relationships:

- **Global Perceptual Network Metaphor (GPNM).** These PNMs embody the relationship between the local host and the network.
- **Entity Perceptual Network Metaphor (EPNM).** These metaphors represent how the local host perceives the remote hosts' network connectivity.

Both GPNM and EPNM will be described in further detail in the next subsections.

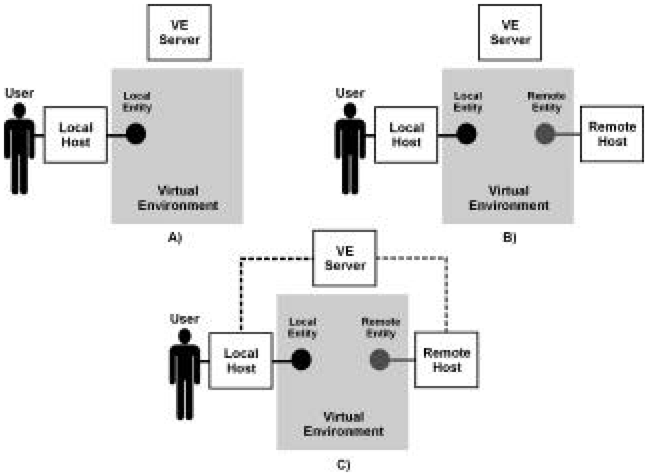


Fig. 3 - Taxonomy of roles and relationships within a virtual environment.
A) The user joins a virtual environment and is alone. B) Another remote user joins the virtual environment in same area. C) The clients communicate with the server to validate their perceived state of the network.

In Fig. 3, a functional step-by-step depiction is given from the perspective of a particular user:

(A) The user enters the virtual environment and while they remain the only existing entity within a particular region, all the PNMs are deactivated (both GPNM and EPNM).

(B) When another remote user enters the same region, the local host activates a dedicated EPNM to monitor the network conditions experienced by the remote host. If the remote user leaves the region, then the associated EPNM is removed.

(C) Each local host assesses its relationship to the network by correlating the data from all the associated PNMs to remote hosts. However, this information requires validation by querying the virtual environment server (or any other network element that may provide the necessary response).

One could dispute the need for EPNMs, since it may be sufficient to have a single global metaphor shared amongst all the users within close spatial proximity of each other. However, the nature of the network forestalls any attempts at triangulation during the duration of a session, since the Internet path properties are not associative, commutative or transitive in nature. Also there is rarely symmetry in paths between two hosts, so even then it would be difficult to have a metaphor mechanism that portrayed the state of the network.

4.1 Entity Perceptual Network Metaphor

For every remote entity that is a source of data traffic, there should be at least one associated Entity Perceptual Network Metaphor (EPNM).

A particular EPNM is triggered whenever the local host perceives that the associated remote entity is experiencing network problems. The particular response of the metaphor depends upon the nature of its implementation.

The design of an EPNM should take into account the particular context of the virtual environment, thus enabling the user to react accordingly.

4.1.1 Probabilistic EPNM

This EPNM is based upon the Heisenberg uncertainty principle from quantum mechanics [32], where the exact position of a remote entity is not known at every instance because of network problems.

The core of the metaphor is a probability model that produces small number of mirror images of the remote entity constrained to an area centered on the last known position. Associated to each image is a density function that reflects the likelihood of where the next position of the entity will be. This will allow the local user to make the necessary assumptions and act accordingly.

The typical application of the EPNM is any virtual environment where it is imperative to know the exact location of entities, such as First Player Shooters (e.g.: Quake™ [22] and Half-Life™ [15]).

4.1.2 Breaking Up EPNM

This EPNM builds upon the culture of mobile phones of loosing speech coherency when a person experiencing bad connectivity. This phenomenon has been commonly labeled as “breaking up”.

The metaphor affects the consistency of the remote entity. As network problems are experienced, the visual representation of the entity is dismantled.

The typical application of the EPNM is Collaborative Virtual Environments (CVE) [10], where the focus is on social interaction. The triggering of the metaphor should prompt the local user to be more socially accommodating of disruptions or communication difficulties experienced by the remote entity.

4.2 Global Perceptual Network Metaphor

The other relationship to consider is between the actual local host and the network. This metaphor affects the user’s interaction with the virtual environment itself, not just a set of remote entities. Taking into account its encompassing nature, this class of metaphors is designated Global Perceptual Network Metaphors (GPNM).

A GPNM is triggered whenever the local host detects that it is experiencing network problems.

There may be either a single GPNM, which incorporates the state of the network, or several GPNMs with each associated to a single characteristic of the network.

The aim of GPNM is not merely to inform the user of the network state, but also to constrain their area of interaction within the virtual environment. The consistency requirements beyond the sphere of influence of the user may be relaxed, considering that the user may not interact.

This allows for the application to easily accommodate the lower QoS provided by the network without degrading the immersive experience of the user.

4.2.1 Weather GPNM

In ancient civilizations, the weather was always seen as the medium that the gods would use to disclose their feelings, ranging from contentment to anger. Although this perception no longer holds true, the various connotations of the weather still remain to this day. Thunderstorms and rain are foreboding, while sunshine with clear skies is auspicious.

Taking advantage of this common association, the Weather GPNM associates bad weather with poor network conditions and good weather with optimal state.

The more severe the network problems, the worse the weather becomes, reducing the area of interaction of the user.

This GPNM is of generic use and may be applied to any application where the virtual environment is open space rather than an enclosed environment.

4.2.2 Light GPNM

Similar to the weather, light (or the absence of it) has similar cognitive connotations. Thus, the lack of light or dark shadows denotes a somber and dread mood.

The Light GPNM uses the intensity of light to reflect the network state. As problems are experienced and intensified, the metaphor starts reducing the ambient lighting. This will leave the virtual environment with pronounced shadows, which reduces the visibility of the user and consequently their area of interaction.

This GPNM is also of generic use but more oriented to applications with virtual environment that are based on enclosed environments rather than open spaces.

5 METAPHOR OPERATIONAL MODEL

The overall operational model of the metaphors is summarized in Fig. 4. The diagram represents the model from the perspective of a particular local host.

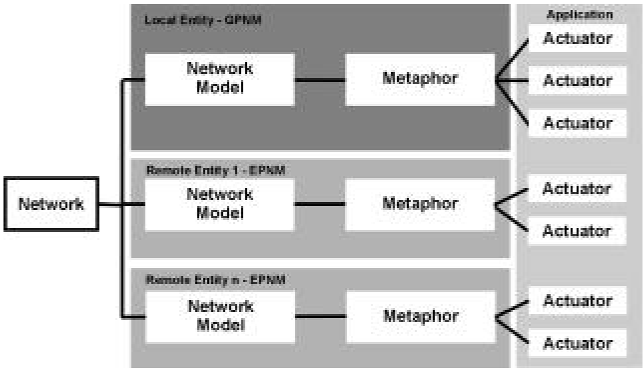


Fig. 4 - Overview of the metaphor operational model

The block diagram presents the three main types of building blocks that are involved, along with their relationships. In an application, there is no restriction on the number of PNM that may be operational. As illustrated, each remote entity may have associated to it one or more EPNMs. In similar fashion, there may be one or more GPNMs active for the local host.

This section focuses on how providing an overview of how PNM work, while the following two sections aid in consolidating the thoughts and understanding by discussing the PNM Manager and a sample prototype application.

5.1 Network Model

As illustrated, each entity and the local host have an instance of a Network Model, which may all be different from one another in the extreme case.

The network model is a simplified model of the underlying network, mapping its current state with a tuple of variables. This model may either represent the local network state or the perceived network state experienced by a remote host.

Each variable represents a network property and their number varies depending on how detailed the particular model is. Although it is possible to have a different network model per entity, in most cases the set of variables will remain the same even if independent instances of the model may exist. Normally, most network models will be a variant of the following tuple (Delay, Loss, Throughput, Jitter).

5.1.1 Delay

The delay denotes the overall latency experienced on the network. The information provided by the network model is of qualitative nature and represents the variance of the actual delay of the data.

5.1.2 Loss

The loss conveys the information regarding the amount of data that is being lost and which never arrives at the various destinations.

The type of data strongly determines its robustness to the existence of loss. The usage of reliability mechanisms may counter the existence of loss by resending the lost data again. However this introduces more delay, which does not fulfill the real-time requirements of a virtual environment.

Even when not considering reliability mechanisms, congestion always introduces additional delay due to the requirement of application data flows to be TCP-friendly as to avoid a congestion collapse in the network [13]. This means that any application must reduce its transmission rate until the network seems to have recovered from the congestion state.

5.1.3 Throughtput

The throughput denotes the rate of the data traffic being sent on the network and correlates to the congestion experienced in the network. When congestion occurs, it is necessary for applications to reduce their throughput and wait until the network has recovered. Thus throughput varies according to the available bandwidth.

The reduction of throughput means that the source is not sending data at the established sampling rate. This leads remote hosts to extrapolate state from less data, which may lead to inconsistencies.

5.1.4 Jitter

The variance in latency is known as jitter. This property affects the smoothing mechanisms that exist to accommodate the data that is arriving erratically.

When the jitter exceeds the threshold permissible by the compensating mechanisms, it is difficult for end-users to assess the state of the network and to adapt to it. In the case of using audio/video streams, the existence of jitter disrupts communication to the point of rendering it incomprehensible [23]. In the case of virtual environments it may invoke motion sickness and causes lapses in consistency of the environment.

5.2 Metaphor

The Metaphor block exists for every PNM, independently of its class being either Entity or Global. This block may be analyzed further according to the diagram in Fig. 5.

Fig. 5 - Internal metaphor operation

5.2.1 Pre-conditions

The network model will contain a current assessment of the network state, however this does not imply that the metaphors should always produce responses. The volatile nature of the network, where the pattern of data traffic is mostly in bursts, requires some filtering and smoothing.

The role of the Pre-Conditions block is to only trigger the metaphor when a specific criterion is matched. This permits for metaphors to be associated to a common Network Model, where customization may be achieved by applying combinatorial or filtering operations to the associated tuple.

5.2.2 Logic

The Logic corresponds to the actual model of the metaphor, determining its operation. The actual model may vary and depends on the application, ranging from probability to keeping track of past states. Considering the highly customized nature of the logic, it is most likely that it will be tightly coupled with the Response block.

5.2.3 Response

The response triggers all the associated Actuators, passing the appropriate parameters as necessary. This design decouples the metaphor from the Actuator, thereby permitting Actuators to be shared across several metaphors. Another advantage of this approach is the possibility of having complex responses by the response of a metaphor activating several Actuators rather than a single one.

5.3 Actuator

The actuator block is responsible for providing the actual feedback of the metaphor to the end user. The actual form of the feedback may be targeted to affect any of the five human senses. There is one actuator per response, but it is possible to combine different actuators to achieve a concerted response triggered by the same metaphor.

The activation of an actuator is done by the mere invocation of its interface, which may contain zero or more parameters. It is convenient to make the interface as generic as possible, to facilitate reusability of the actuator with different metaphors.

In Mayhem [20], the actuators were either absent of parameters or included a single parameter to denote the intensity of the response. Although not used, another parameter foreseen is the frequency for responses with periodic behavior.

6 PNM MANAGER ARCHITECTURE

The PNM Manager is the element responsible within an application to assess the state of network, thus providing the various EPNM with the necessary information to build their models of the network. The management of the various entity based and global metaphors is also the responsibility of the manager. The block diagram of Fig. 6 illustrates the internal architecture of the PNM Manager.

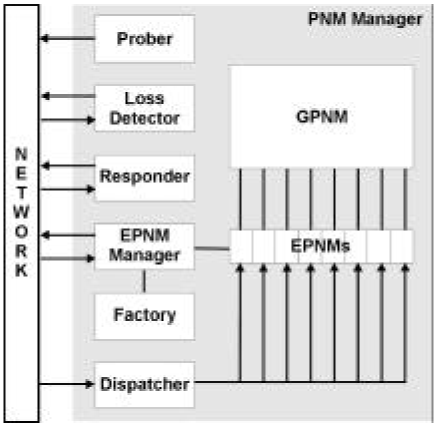


Fig. 6- PNM Manager building blocks

The PNM represent, by means of metaphors, the state of the network. Towards this end it is necessary to have in place mechanisms to assess the network with minimal cost in terms of performance and traffic overhead, whilst maximizing flexibility.

Although it would be possible to discern certain properties of the network from the data traffic generated from the application, such an approach would make it highly inflexible with low reusability. Therefore, the adopted approach was to introduce an independent set of messages for network assessment.

In all cases, the generic communication model considered is many-to-many, independently if the underlying transport protocol is multicast or some other group communication infrastructure. This choice influenced the design and operation of the PNM Manager, since traffic implosion was to be avoided and scalability a requirement.

With the aim of providing maximum flexibility, the PNM protocol follows the principles proposed by composeable protocol stacks [Bhatti], where the aim is to build protocols from smaller components with their functionality defined depending upon the granularity of their design. Therefore, the paper presents the necessary fields within the messages but without establishing the order or format. This approach should allow the protocol to be either independent or integrated into the remainder application protocol.

All the messages of the protocol have in common a field to identify the type of message and a unique global identifier that identifies the sender. Considering a network model that comprises of the tuple (latency, loss, jitter), the remainder of the section will describe how the diagram block of the PNM Manager operates and interacts with the EPNM/GPNM. The network throughput is a difficult property to measure, requiring stable monitoring which in the case of a virtual environment with its volatile nature seems difficult to achieve.

6.1 Latency

The measurement of network latency, and to a certain extent loss, is achieved by the use of the pair *Prober*/*Dispatcher* on the sender side and the *Responder* on the receivers.

The *Prober* periodically sends to the group a Probe, which includes the timestamp of when it is sent along with a sequence number. When a remote host receives a Probe, it will respond with a *ProbeResponse*, which includes the timestamp of the reply and the identifier of the receiver along with most of the information stripped from the *Probe*. The inclusion of the sequence number and sender identifier allows the appropriate host in the group to recognize the response and correlate it to the original *Probe*.

The responses are processed by the *Dispatcher*, which derives the latency from the available timing information contained in the *ProbeResponse*. The information is forwarded to the appropriate EPNM.

When a *Probe* is sent, a timer is set to RTT_{max}. When the timer expires, the *Prober* queries the *Dispatcher* which remote hosts neglected to reply within the expected time. To each of those that failed to reply, the *Prober* sends a *ProbeReplyRequest,* which obliges the target recipient to reply with a *ProbeResponse*. Associated to each sent *ProbeReplyRequest* is a dedicated timer. The expiration of this timer increases the sequence number of the request and a new timestamp is issued when resending. The request message is sent repeatedly until either a reply is received or a configured threshold is exceeded. In the case the threshold is exceeded, then the *Dispatcher* assumes that the remote host has experienced a bizantine failure and is unable to recover. In this case, the *LossDetector* is notified with a *FailureEvent* indicating the identifier of the remote host in question.

6.2 Loss

The periodicity of the *Probes* may not suffice to determine the existence of loss in the network, although latency and loss are correlated. To address this issue, the protocol includes a passive mechanism similar to the packet burst mechanism in the receiver based multicast congestion control based on layers []. So the sender periodically sends to the group a series of *Alive* messages containing just a sequence number.

None of the receivers will reply to an *Alive* message, but will discern if any messages of the burst have been lost and how many. Depending on the rules of the EPNM associated to the particular sender, the loss detected may trigger the metaphor or not.

All the receivers are equally configured to recognize the size of a burst as to how many messages are involved. Thus, the format of the *Alive* message is simple with just the type field and the global identifier of the sender.

This approach isolates the sender from any responsibility towards the receivers. The span of an epoch is determined based on two fixed parameters, one establishing the age and the other the size. The former determines the time after which all *Alive* messages belonging to the burst are ignored, while the latter determines how many messages belong to the burst. When a message fails to respect either or both parameters, then it is considered a loss.

To avoid a surge in control traffic, inadvertently causing loss due to simultaneous bursts, the receivers adopt a suppression mechanism. Therefore, upon receiving an *Alive* message, the associated local timer for sending the burst is reset.

6.3 Jitter

This network characteristic is measured by the analysis of the variation in the delay observed when receiving both the *Probe* and *Alive* messages from a particular source.

6.4 Announce/Leave

At any given instance, the EPNM Manager is responsible for managing the set of EPNM representing the corresponding remote entities that are of interest to the local host.

The set of EPNM neither is determined at the beginning of a session or does it remain static throughout the duration of the session. The dynamics of the set enforces the need of a mechanism to inform the Manager when modifications are necessary. The adopted mechanism is based on the inclusion of two simple messages to the PNM protocol, indicating when an EPNM is necessary or when it becomes obsolete. Both messages merely contain the field defining its type and the global unique identifier of the host making the announcement.

The modifications to the set of EPNM is tightly coupled with the operation of the Area Of Interest Management (AOIM) and consequently whenever a host verifies a change in its area of interest, then either an *Announce* or *Leave* message is sent out.

When the local host joins a new area of interest, then it sends out an *Announce* message to make their presence known.

The receiving hosts upon reception of the message will determine if an EPNM already is place for the particular remote host. Should this be the case, then the message is ignored, otherwise the appropriate EPNMs are created to represent the announcing host and in response the local host will also send *Announce* messages.

Analogous to a host announcing its presence, there exists a corresponding operation for when host leaves the previous area of interest. In this case, the host sends a *Leave* message before leaving the current active area of interest. When the receiving hosts retrieve the message, their EPNM Manager removes from the set of EPMNs the one corresponding to the particular remote host.

For efficiency purposes, a cache mechanism should be employed to avoid unnecessary instantiation of EPMNs.

7 IMPLEMENTATION

The PNM architecture has been implemented as one of the component frameworks within the Mayhem project [20]. As part of the project there exists an online game prototype to demonstrate the use of the existing components. The game consists of a Massive Online Role Playing Game (MORPG) where a user is embodied as an avatar that may interact with other users or the environment itself. The current version is based on a 2D tile-rendering engine.

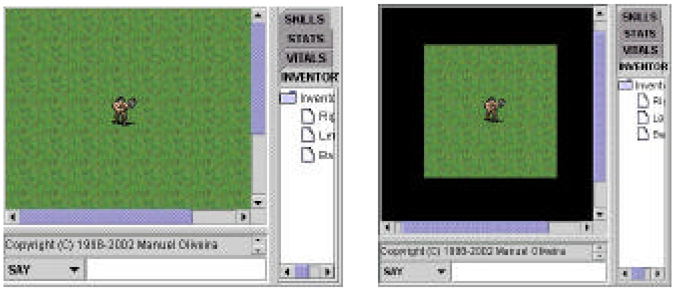


Fig. 7 - The GPNM without and with moderate packet loss

Every EPN can be implemented as an extension to the component framework. For the purpose of the prototype, a single plug-in was developed for each metaphor class:

- **Passing Out (GPNM).** This metaphor (Fig.6) associates the network state to the local visibility of the user. The number of visible tiles is determined by the network problems experienced. The worse the state, the fewer tiles are visible around the avatar. Ultimately, the user passes out and the screen becomes black.
- **Transparency (EPNM).** This metaphor is associated to each remote avatar. If the local host perceives that the remote host is experiencing problems, then the respective avatar will be rendered with a composite layer with appropriate values of alpha channel.

To assess the network model, a control protocol was developed that collected information from remote hosts via a dedicated multicast address. The server would also be probed for validation in the case before changing the GPNM.

8 DISCUSSION

This paper has presented the PNM as a mechanism to explicitly present to the user the current state of the network. Whenever the network experiences limitations that ultimately may affect the experience of the user's mutual interaction, the metaphors are activated and reflect the intensity of those limitations. However, the metaphors do not necessarily need to be tightly coupled solely with network conditions, for hardware and software limitations of the hosts could equally contribute to the metaphors.

Most of the stuff could be discerned from the current network mechanisms to counter congestion or any other with monitoring mechanisms.

Talk about the use of policy in the case of two solutions. One is the use of metaphors. Best effort replies on the application of the metaphors and social understanding, the use of policies enforce social structuring.

9 CONCLUSIONS AND FUTURE WORK

Although not widely recognized, the end-users of today are increasingly more aware of the role of the network in multi-user applications over the Internet. The analysis of a survey [29], targeted to online game players, strongly suggests that users are aware of the impact of the network and would like to have more information of its state. This result has been found in small experimental studies [5, 30].

The use of Perceptual Network Metaphors (PNM) will provide user information regarding the network state, but in an integrated fashion so as not to detract from their experience. At the moment, the metaphors have been implemented as a component framework within the Mayhem project. A simple 2D MORPG game engine was implemented to demonstrate the operation and functionality of the metaphors. The next development stage will consist of developing Actuators for 3D rendering and to implement more engaging visual metaphors.

The usage of PNMs should allow the user to integrate into their perceptual model the notion that within a virtual environment the propagation delay of information is constrained by a distance that correlates the virtual spatial one with the actual network topology. The next stage will involve an extensive user trial to evaluate the perceived impact of network state on users' experience with and without the use of PNM.

Although the metaphors discussed and actually implemented are targeted to the visual senses, it is possible to associate an Actuator to any human sense, ranging from haptic to auditory. Part of the future work is the study of alternate forms of Actuators, targeting different human senses.

Acknowledgements

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

References

1. Arlitt, M. and Williamson, C., “Web Server Workload Characterization: The Search for Invariants”, Proc. ACM SIGMETRICS’96, USA, 1996
2. Bailey, R., “Human Performance Engineering – Using Human Factors/Ergonomics to Achieve Computer System Usability”, Prentice Hall, NJ, 2nd Edition, 1989
3. Bernier, Y., “Latency Compensating Methods in Client/Server In-Game Protocol Design and Optimization”, Proc. Game Developers Conference, San Jose, March 2001
4. Bhatti, N. and Schlichting, R., “A System for Constructing Configurable High-Level Protocols”, ACM SIGCOMM’95, August 1995
5. Biocca, F., “The Cyborg’s Dilemma: Progressive Embodiment in Virtual Environments ”, Journal of Computer Mediated Communication, Vol. 3, N. 2, September 1997
6. Bouch, A., Kuchinsky, A. and Bhatti, N., “Quality is in the Eye of the Beholder: Meeting Users’ Requirements for Internet Quality of Service”, Proc. ACM CHI’2000, Netherlands, 2000
7. Braden, R., Clark, D. and Shenker, S., “Integrated Services in the Internet Architecture: An Overview”, IETF RFC, July 1994
8. Briscoe, R., and Bagnall, P., “Taxonomy of Communication Requirements for Large-Scale Multicast Applications”, IETF Internet Draft, July 1997.
9. Chassot, C., Lozes, A., Garcia, F., Diaz, M., Dairaine, L. and Rojas, L., “Resource reSerVation Management Architecture for DIS-like Applications”, LAAS/CNRS Technical Report N. 99021, January 1999
10. Cheshire, S., “Latency and the Quest for Interactivity”, November 1996. rescomp.standford.edu/~cheshire/
11. Churchill, E. and Snowdon, D., “Collaborative Virtual Environments: An Introductory Review”, Virtual Reality. Research Developments and Applications. Vol. 3, 1998
12. Conner, B. and Holden, L., “Providing a Low Latency Experience in a High Latency Application”, Proc. ACM Interactive 3D Graphics’97, Providence, 1997
13. Crovella, M. and Bestavros, A., “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, IEEE/ACM Transactions on Networking, Vol. 5, N. 6, December, 1997
14. Diot, C. and Gautier, L., “A Distributed Architecture for Multiplayer Applications on the Internet”, IEEE Networks Magazine, Vol. 13, N. 4, July-August, 1999
15. Feldmann, A., Gilbert, A., Huang, P. and Willinger, W., “Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control”, Proc. ACM SIGCOMM’99, Cambridge, 1999
16. Floyd, S. and Fall, K., “Promoting the Use of End-to-End Congestion Control in the Internet”, IEEE/ACM Transactions On Networking, Vol 7, N. 4, August 1999
17. Fraser, M., Glover, T., Vaghi, I., Benford, S., Greenhalgh, C., Hindmarsh, J. and Heath, C., “Revealing the Realities of Collaborative Virtual Reality”, Proc. ACM CVE’00, San Francisco, 2000
18. Half-Life: http://www.valvesoftware.com/
19. Harvey, W., “The Future of Internet Games”, Position Paper, Modeling and Simulation: Linking Entertainment and Defense, National Academy Press, Washington DC, 1997
20. Henderson, T., “Latency and User Behaviour on a Multiplayer Game Server”, Proc. ACM NGC 2001, London, November 2001
21. Matijasevic, M., Gracanin, D., Valavanis, K. and Lovrek, I., “Towards QoS Specification for Distributed Virtual Environments’, Proc. IEEE 10th MEleCon 2000, Vol. I, 2000
22. Mauve, M., “How to Keep a Dead Man from Shooting”, Proc. 7th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services’00, Netherlands, October, 2000
23. Citation withheld
24. Park, K. and Kenyon, R., “Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment”, Proc. IEEE VR’99, Houston, March, 1999
25. Paxson, V. and Floyd, S., “Wide-Area Traffic: The Failure of Poisson Modeling”, Proc. ACM SIGCOMM’94, London, 1994
26. Quake: http://www.idsoftware.com/
27. Ramjee, R., Kurose, J., Towsley, D. and Shulzrinne, H., “Adaptive Playout Mechanisms for Packetised Audio Applications in Wide-Area Networks”, Proc. IEEE Infocomm’94, Toronto, April, 1994
28. Ryan, M. and Sharkey, “Causal Volumes in Distributed Virtual Reality”, Proc. IEEE Computational Cybernetics and Simulation, Vol. 2, 1997
29. Saltzer, J., Reed, D. and Clark,D., “End-To-End Arguments in System Design”, ACM Transactions on Computer Systems, Vol. 2, N. 4, November, 1984
30. Schroeder, R., Steed, A., Axelsson, A., Heldal, I., Abelin, A., Wideström, J., Nilsson, A. and Slater, M., “Collaborating in networked immersive spaces: as good as being there together?”, Computer & Graphics, Elsevier Science Ltd, Vol. 25, N. 5, October 2001
31. Sharkey, P., Ryan, D. and Roberts, D., “A Local Perception Filter for Distributed Virtual Environments”, IEEE Proc. Virtual Reality Annual International Symposium, 1998
32. Shenker, S., Partridge, C and Guerin, R., “Specification of Guaranteed Quality of Service”, IETF RFC 2212, 1997
33. Singhal, S. and Cheriton, D., “Exploiting Position History for Efficient Remote Rendering in Networked Virtual Reality”, PRESENCE: Teleoperators and Virtual Environments, Vol. 4, N.2, Spring, 1995
34. survey: url withheld
35. Vaghi, I., Greenhalgh, C. and Benford, S., “Coping with Inconsistency due to Network Delays in Collaborative Virtual Environments”, Proc. ACM VRST’99, London, December, 1999
36. Vicisano, L., Rizzo, L. and Crowcroft, J., “TCP-Like Congestion Control for Layered Multicast Data Transfer”, Proc. IEEE Infocomm’98, 1998
37. Vogel, J. and Mauve, M., “Consistency Control for Distributed Interactive Media”, Proc. ACM Multimedia 2001, Ottawa, October, 2001
38. Wheeler, J. and Zurek, H., “Quantum Theory and Measurement”, Princeton University Press, 1983
39. White, P., “RSVP and Integrated Services in the Internet: A Tutorial”, IEEE Communications Magazine, May 1997
40. Wroclawski, J., “Specification of the Controlled Load Network Element Service”, IETF RFC 2211, 1997
41. Yang, J. and Lee, D., “Scalable Prediction Based Concurrency Control for Distributed Virtual Environments”, Proc. IEEE Virtual Reality 2000, New Brunswick, March, 2000
42. Yu, H., Zhou, Q., Makrakis, D., Georganas, N and Petriu, E., “Quality of Service Suppot of Distributed Interactive Virtual Environment Applications in IP Networks”, Proc. IEEE Virtual Reality 2001, Yokohama, March, 2001

10 APPENDIX

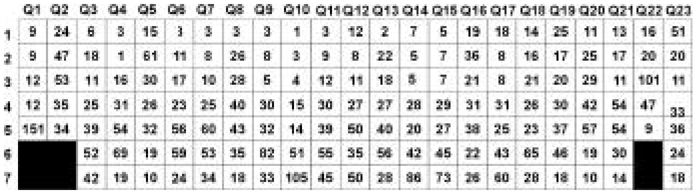
10.1 APENDIX A

The questionnaire used for the online survey is based on 23 questions with responses being scaled from one to seven. The intention was not to obtain quantitative replies from the respondents, but rather their qualitative perception regarding the issue being addressed in the question.

1. For how long have you played online games?
2. On average, how many hours a week do you play online games?
3. How much do games influence your purchases of new computer hardware?
4. Overall, how proficient are you as a player?
5. When you are playing a game, to what extent are you aware of your surroundings (i.e.: world outside the computer)?
6. How much do you have a sense of being in the game world?
7. Do you have a sense of being in the same space with other players?
8. How often do you notice disruptions in the game (excluding external disruptions such as telephone, people interrupting, etc)?
9. What proportion of game disruptions do you think are due to network problems as opposed to software problems?
10. How annoying are game disturbances that result from network problems?
11. Do you become more aware of your physical surroundings when network problems occur?
12. Do network problems disrupt your sense of being in the same space with other players?
13. When you abandon a game, how often are network problems the main cause?
14. How significant are ping times in choosing a game server?
15. How annoying is it when you have a much higher ping time than other players?
16. How often do you check your ping time (status) during a game?
17. Do you prefer servers where everyone has similar ping times to you?
18. How often do you play games with people you already know?
19. How often do you meet new players in games and play with them in future sessions?
20. Can you adjust your game play in the presence of network problems?
21. Does learning to anticipate network problems affect your game play?
22. When network problems occur, how would you prefer to know about them?
23. Would you be willing to pay (even a small amount) for a service that reduced network problems in games?

10.2 APENDIX B

The following table summarizes the 193 responses obtained from the online survey conducted through the period of three months between September-November 2001. A seven scale response was used to accommodate the range from a weak to strong response.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23
1	9	24	6	3	15	1	3	3	3	1	3	12	2	7	5	19	18	14	25	11	13	16	51
2	9	47	18	1	61	11	8	26	8	3	9	8	22	5	7	36	8	16	17	25	17	20	20
3	12	53	11	16	30	17	10	28	5	4	12	11	18	5	7	21	8	21	20	29	11	101	11
4	12	35	25	31	26	23	25	40	30	15	30	27	27	28	29	31	31	26	30	42	54	47	33
5	151	34	39	54	32	56	60	43	32	14	39	50	40	20	27	38	25	23	37	57	54	9	36
6			52	69	19	59	53	35	82	51	55	35	56	42	45	22	43	65	46	19	30		24
7			42	19	10	24	34	18	33	105	45	50	28	86	73	26	60	28	18	10	14		18