

多人虛擬環境中以使用者感受為導向之網路服務

User-centric Network Service in a Multi-User Virtual Environment

周承良¹

李蔡彥²

李秀惠¹

1 國立台灣大學資訊工程學系

電話: +886-2-23630231 Ext. 3235-308

E-mail: {hh_lee,b7506011}@csie.ntu.edu.tw

2 國立政治大學資訊科學系

電話: +886-2-29393091 Ext. 63379

E-mail: li@cs.nccu.edu.tw

摘要

在多人虛擬環境中縮減網路訊息傳輸量一直是許多研究關心的議題，而本篇論文從「使用者感受」的角度來檢視在此環境中如何篩選不必要的訊息傳遞，目標是盡量讓系統當中的使用者滿意。在本篇論文中將會提出一套評估使用者滿意度的基本模型，模型中評估滿意度的指標是以使用者在系統中活動的狀態以及其硬體的能力為參考。我們依據此模型也實作了一套控制系統以減少網路訊息傳輸量，並且在不同的環境下對此系統做實驗及觀察。最後我們會針對實驗數據作分析和討論，然後提出一些現有的問題及未來的工作。

關鍵詞：多人虛擬環境、網際網路應用、網路服務品質

Abstract

Multi-User Virtual Environment (MUVE) is a popular topic attracting much research focus in the recent years. Reducing the amount of information transmitted over the network for such an environment is a crucial task for good system performance. This paper focuses on how to screen unnecessary information transmission in such a system from a user's point of view. We will try to satisfy as many users as possible with a just-enough strategy. We propose a set of network service models to assess user's satisfaction according to the user's activity and hardware capability. We have implemented a control system with these models to reduce the amount of network transmission and conducted experiments to observe the system performances under different experimental settings. Finally, we analyze the results of the experimental data and discuss them with our possible future extensions.

Keywords: Multi-User Virtual Environment, Internet Application, Network service quality.

1. 前言

近年來電腦遊戲蓬勃發展，而網路遊戲更是其中異軍突起的一項新興產業。網路遊戲以系統的角度來分析，

可以說多人虛擬實境系統(Multi-User Virtual Environment, MUVE)[1]的一種，本篇論文即是著眼於此類系統的研究。

多人虛擬環境系統可以應用的地方越來越廣泛；舉凡任何人與人之間在現實生活中的互動模式，幾乎都可以透過網路並經由虛擬環境營造出來。另外，在虛擬環境中，可以模擬許多現實生活中難以達成的互動方式；例如現在市面上最常見到的線上遊戲，就是很成功的應用實例，其他諸如軍事模擬、電子會議等方面的應用，也都可見到成功的案例。但是在這類系統當中，都存在著一個主要的問題：如何傳遞大量的訊息。此類系統的瓶頸多半出現在網路傳輸的部分，因此也有許多的相關研究致力於降低資料傳輸量。雖然目前的系統已經有許多降低資料傳輸量的方法，但是大都是以一視同仁的方式來服務所有的使用者。另一方面，網路服務的研究，雖有許多網路服務品質保證(QoS)的方法，但多從網路層著手，較少有針對應用層特性(特別是虛擬環境系統的應用)加以客製化的。有鑑於此，我們對於是否所有使用者都需要等等的服務品質這個問題提出質疑，並且站在使用者感受的角度來分析這個問題。在本論文中，我們將試著從「使用者感受」的角度來篩選傳輸的訊息數量，企圖藉此增加系統的可用性。另外，我們預定採用的系統架構是主從式(client-server)架構，但研究成果的運用並不侷限於此種架構。

我們設計的系統希望能針對使用者的差異給予其所需要的網路服務品質，並依照 server 當時的狀態調整網路服務品質。而我們所希望達成的目標是盡量讓所有的使用者對系統服務滿意，並且避免 server 端不必要的服務資源浪費。

本篇論文的主要貢獻在於：

- 提出以使用者感受為角度來調整在多人虛擬環境系統下網路服務品質的概念。
- 提出一套評估使用者感受的滿意度基本模型。
- 實作一套應用此概念的控制系統，並以實驗佐證及探討此概念和作法的優缺點。

2. 相關研究

我們在本節將介紹多人虛擬環境系統的大致

架構，以及目前已經存在的一些系統，並且探討經常遇到的問題。

2.1 多人虛擬環境系統分析介紹

一般的多人虛擬環境系統大致上可以分成以下幾部分：「邏輯運算部分」、「通訊連線部分」、「輸出輸入裝置」和「虛擬世界描述語言」。

通訊連線部分是負責整個系統所有參與者資料的傳輸，互動的結果幾乎都是掌握在這個部分。互動過程當中資料傳輸的模式、傳輸的內容、傳輸內容的正確性等，都會對於整個系統表現造成很大的影響，因此這部分也可以算是多人虛擬環境系統議題最多的一環，本篇論文鎖定的方向也是與這部分有關，因此以下我們將就這個方向做比較詳盡的介紹和討論。

2.2 多人虛擬環境系統中的通訊連線

通訊連線的部分可以在分成兩個主要部分：「整體連線架構模型」和「網路傳輸相關」，我們分別在下面討論其內容和相關的問題。

連線架構模型大致上可分成三類：*client-server* 架構、*peer-to-peer* 架構、特殊變種架構。*client-server* 是最常見的架構，所有訊息由 server 端控管，架構簡單清晰。*peer-to-peer* 則比較複雜，每個端點之間都會彼此傳遞訊息[2]。而變種的架構則是比較新型態的架構，例如多層 server 的架構，或是混合前面兩種類型的架構。由於我們改進的方向不在於調整整體架構，因此將會採用實做較簡易且最多人使用的 *client-server* 架構來設計我們的系統。

通訊相關的方面則有以下這些議題：訊息傳遞形式、延遲、訊息可靠性、網路頻寬。訊息傳遞的方式，大致上可分成 broadcast、multicast 和 unicast。而我們的系統則是採用 unicast 的方式來連線，原因在於其單純直接的特性，並且在 WAN 的環境下實現較無障礙。延遲方面的問題則也是相當大的挑戰，它會造成整個系統中以時間軸的角度來觀察使用者認知時，會出現不一致的情況。而其中延遲最嚴重的部分往往是訊息在網路上傳遞所耗費的時間，因此這方面的研究多半朝兩個方向進行，一個是實質上的改善延遲的時間，另一個則是減少使用者對於延遲的敏感度。訊息可靠性的目的在確保能準確的發送訊息至接收端。網路的頻寬則是決定一定時間內最大的訊息傳輸量；頻寬不足的話，就會造成網路壅塞的現象。

2.3 通訊連線相關問題的解決方案

我們在這一小節根據前述通訊連線所遇到的問題陳述現有的解決方式。這方面問題的解答，大致上可分成兩類：「減少訊息傳輸量」和「分散系統負擔」。

在減少訊息傳輸量的方面，有以下幾種典型的方式：「動態路徑預測」、「視角篩選」、「虛擬世界

分割」、「資料精細度調整」及「興趣導向」。動態路徑預測(Dead Reckoning[11])是個已被廣為應用的技術。許多相關研究都是在改善預測的準確性或是運算的效能[15]。視角篩選則是根據使用者所能看到的範圍來篩選訊息，而 RING[6]正是採用這個方法。也有一些研究是關於降低視角篩選時所耗費的時間，例如[4]提出了一套更有效率的排序演算法來加快視角篩選的工作。虛擬世界分割則很常用在 peer-to-peer 的架構下，在[11]這篇當中曾提出幾種分割的類型。目前最廣為採用的是空間分割法，將整個虛擬世界分割成好幾個區域，利用區域來篩選訊息或是更進一步的將不同區域的訊息處理分配給不同的機器。另外在 ATLAS[9]、[2]、[14]、[5]等研究中也都會應用虛擬世界分割的概念。資料精細度的調整(Level of Details, LOD)最典型的就是模型多重解析度的運用[7]。興趣導向則是指只傳遞使用者有興趣的資料，這部分在[13]、[2]當中也有些描述。另外在[9]以及[10]當中，他們也提到將使用者活動的方式納入興趣模型當中，以使用者與某類物件的互動頻率，來判斷他的興趣，這種概念與我們這篇論文的觀念類似。上面所提到的方法，雖然都能有效的降低系統訊息的傳輸量，但是除了興趣導向有針對使用者的差異調整內容之外，其餘調整方式都是給予所有使用者相同的篩選方式，並沒有考量到個別使用者的感受和需求。

在分散系統的負擔這方面的方法上，則有以下幾種新型架構的出現：「*client-server* 和 *peer-to-peer* 綜合體」及「多 server 架構」。在[2]當中規劃了一種複數 server 的架構，server 之間以快速的網路連接傳遞訊息，而每個 server 所負責的範圍是由虛擬世界去切割分配。在[14]當中則是提出一種動態分割虛擬世界，並且從處於此塊分割的使用者當中挑選一位擔任管理者，藉此分擔 server 的工作，簡單的說就是自動的挑選一些使用者的機器來擔任區域 server。在[9]當中提出一套系統，其主體架構是 *client-server*，但是部分的資料也可透過 client 之間以 multicast 的方式傳遞，藉此分擔 server 的負擔。

而本篇論文並不探討以多個 server 分散處理的方式，我們專注在只有單一台 server 的情況下，該如何使系統有更高的可用性。研究的成果也可以延伸到多個 server 的系統架構下。

3. 問題探討與系統規劃

3.1 理想系統的探討

一個多人虛擬環境系統幾乎不可能達到完全理想的境界，因此我們嘗試著就使用者感受的角度來思考一個系統的問題：如何能使一個不完美的系統依舊能讓大多數的使用者接受並且滿意？

既然是以使用者滿意度而不是系統完美為目標，因此我們首先必須思考滿意度的評估方式。而很明顯的，滿意與否是主觀性的答案，勢必因著使用者本身的一些狀態以及使用系統所需要付出的

代價而有所不同；也因此要讓所有使用者感受到滿意，未必需要給予等同的客觀條件。所以我們嘗試著從使用者的異質性來思考：使用者主觀性感受和客觀性環境條件之間的關聯，並企圖從這關聯性當中尋找讓使用者滿意的可能方式。

3.2 滿意度與服務品質之關聯性及服務品質調整方式

3.2.1 探討滿意度與系統所給予服務之關聯性

既然我們系統的目標是讓使用者滿意，而現有的系統卻都是採用一視同仁的方式來服務所有的使用者，於是在這裡我們提出一項質疑：是否所有使用者都需要相同品質的服務才會感覺滿意？

不可否認的，所有的使用者都會希望自己所得到的服務是最好的，但他們「所得到的服務是否能完全反應在他們的感受上」這個議題就很難下定論了。又除了使用者主觀的感受之外，「客觀的軟體環境限制是否能反應出系統所給予的服務品質」也是個值得深思的問題。

我們認為使用者的滿意度未必正比於系統所給予的服務品質，還是還會受到使用者本身狀態的影響。因此，關於服務品質的定義，我們在此提出幾項假設原則：

- 服務品質接近使用者客觀的硬體能力將能讓使用者滿意。
- 服務品質與使用者對系統專注度成正比將能讓使用者滿意。
- 服務品質根據使用者的互動類型調整將能讓使用者滿意。

根據這些假設，我們嘗試定出一套使用者滿意度的指標，用以衡量系統資源的分配方式。並且我們把服務品質的焦點放在多人互動部分的網路服務上，來探討此類服務跟使用者滿意度的關聯性。所以，系統服務品質此後都泛指與網路服務相關的品質。

我們透過應用程式的設計，在網路應用層進行資料傳輸量的調整，以控制整個系統的服務狀態。我們對於系統提供的互動模式，目前暫時先限制在簡單的移動，說話等方式，因為簡單的互動類型能讓我們比較容易探討使用者的感受與其活動類型之關聯性。

即使我們針對使用者的差異盡量提供合理的服務，server 端還是必須面臨資源耗盡的一天。因此在 server 負擔過重的時候，我們會適度將使用者的服務降級以求 server 的持續運作。並且在資源充裕的時候自行提升服務品質。

綜言之，我們希望能配合 server 的狀態，考量使用者的滿意程度和能力，來調整我們系統整體的服務，以求 server 資源的最佳利用，並且能在某些情況下紓解 server 的效能瓶頸。

3.2.2 思考服務品質調整方式

依前所述，我們先將問題分成兩個部分：第一個部分是觀察系統的狀態，第二個部分則是針對這些狀態，找出我們所能夠掌握的差別服務項目，並且給予使用者合適的服務。

在觀察狀態部分，我們觀察對象包括 client 與 server。就觀察方式而論，由於我們既定的策略是由應用程式層來做控管，因此在發送端跟接收端均將藉由應用程式來監測這些狀態資料。就觀察內容而論，server 端可以觀察的狀態大致上分成「邏輯運算負載」和「IO 負載」。以多人虛擬環境系統的 server 為例，這兩項的狀態已能完整的表示 server 的狀況。而 client 端可觀察的狀態則可歸納為「顯像能力」、「IO 能力」、「使用者活動頻率」及「使用者活動類型分佈」等。這幾個項目是根據我們之前提到能讓使用者滿意的假設為前提所提出的。前兩項是用以表示 client 端的硬體能力；後面兩項則是用以評估使用者主觀狀態的項目。我們認為這兩項資訊跟使用者對於系統的專注程度有很大的相關性。

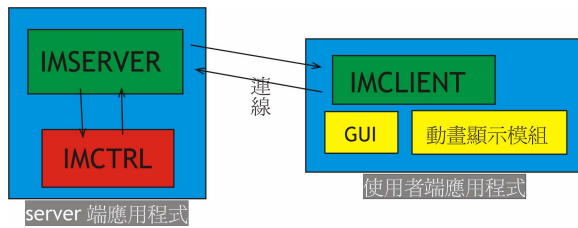
我們所能掌握的服務項目最主要分成「服務頻率」和「服務內容」兩方面。在此系統架構中，服務頻率所代表的是單位時間內傳送資料的次數，這會影響到使用者對於整個虛擬環境中各狀態正確性的掌握，進而影響他與環境互動的反應。在服務內容方面，大致上可以包含「動態路徑預測」、「勢力範圍篩選」和「重要性篩選」等幾項技術。動態路徑預測已經是個相當普遍的技術，但是我們期望能給予每個使用者不同的誤差容忍；另外我們也希望給予使用者有不同大小的勢力範圍，經由這個範圍來控制他所獲得的訊息數量；最後我們希望給予每個使用者，甚至是環境中的每個物件有個重要性的指標，藉由它繼續對訊息作篩選的工作。

整體而言，我們的系統的規劃方向是以前四項 client 端的狀態來針對服務頻率和服務內容做些差異性的調整，而調整的方針則依據前面我們提出的原則；也就是說在給定合理的初始值後，根據後兩項 server 的狀態來作整體或是個別性的降級和升級，目標是讓 server 端能保持在一定運作水準的前提下，提供使用者較為合理且滿意的服務。而系統狀態和服務項目調整之間詳細的關連性，我們在後面會詳加介紹。

3.3 規劃一套能針對使用者感受和能力以及 server 的現況提供不同品質服務的系統

基於跨平台的因素考量，我們實作的語言是 Java。Client 端是以 java applet 的形式，讓使用者在無須安裝額外程式的情況下就能使用本系統。根據之前我們所討論的，我們可以把整個系統劃分成兩個部分：基本的多人虛擬環境系統及監控偵測處理的模組。

關於第一個部分的虛擬環境系統，我們所使用的是「IMNET 多人虛擬環境系統」[16]。我們整個系統是以 IMNET 為底層架構，再加上我們自己的監控偵測處理模組：IMCTRL。有關 IMNET 的粗略



圖一、IMNET 多人虛擬環境系統架構圖

架構及與 IMCTRL 的關係請參見圖一。其中，我們在 server 端插入我們即將要介紹的 IMCTRL，負責處理資料的篩選和分配、server 和 client 狀態的偵測以及滿意度的評估。

根據我們前一小節的討論，我們可以歸納出幾項我們在 IMCTRL 模組所需要的功能：

- 定期偵測 server 端和 client 端的狀態
- 根據狀態評估我們所預計給予的網路服務
- 實現我們給予的服務

根據我們需要的功能，以下我們條列出我們所規劃的模組，其架構如圖二所示：

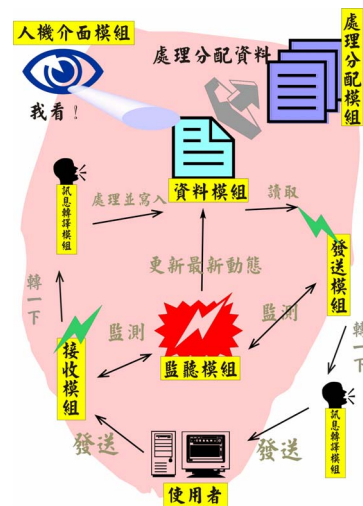
- 資料模組：負責儲存整個 server 所有的資料。
- 接收模組：負責從使用者端接收資料。
- 發送模組：負責從資料模組中取得並發送資料給使用者。
- 監聽模組：定時監聽 server 和 client 中我們之前所提到需要用來評估的各項狀態。
- 處理分配模組：根據資料模組中的狀態資料篩選所需要發送的訊息。
- 測試者觀察調整人機介面：主要是讓我們測試的時候有良好的介面可以方便調整各項參數。
- 訊息轉譯模組：轉換 IMNET 的訊息。

4. 滿意度模型、服務調整策略及公式

4.1 滿意度模型的制定

我們將進一步根據前面的假設，描述如何依據使用者的狀態來訂定讓使用者滿意的服務品質。

在使用者的狀態方面，觀測模組觀測得知的使用者端的個別狀態包含顯像能力(Frame Per Second, FPS)、IO 能力(throughput)、活動頻率、及活動類型等。顯像能力目前由 client 端動畫模組提供更新一個畫面所需時間的資料。關於 IO 能力的測量，我們希望能夠知道 client 端接收每一封訊息所需要耗費的時間；粗略的測量的方式為送出兩封大小不一的訊息，比較他們接收所需要時間的差來判斷我們所需要的資訊。舉例來說，假設 A 訊息大小 10MB，B 訊息大小 5MB，A 訊息從 server 發送到 client 接收完畢耗時 x 秒、B 訊息耗時 y 秒，則 x-y 會當成傳送 5MB 所需要耗費的時間。使用者活動頻率的測量則是非常的直觀簡易，我們只要累計 client 端發送的訊息次數即可。使用者活動類型分佈同樣也是累計 client 端發送的訊息次數，跟前項所不同的是採分類累計。



圖二、IMCTRL 的架構圖

而在改變使用者滿意的服務品質方面，系統所能調整的參數包含以下項目：

1. **重要度(包括自己和他人)**：「自身重要性」這個參數在目前主要是表示使用者動態行為對他人的重要性；而「他人重要性」則是表示使用者本身所需要接收的動態行為資料之所有者其「自身重要性」的最小值。(舉例來說，當 A 的他人重要性 \geq B 的自身重要性時，A 才會收到 B 的動態行為資料。)這個參數其實還可以作更廣泛的應用，若是虛擬環境當中每個物件都有重要度的欄位，則根據使用者與其互動的頻率，也可以量化此物件在使用者心中的重要性，藉以篩選訊息。不過在此我們暫不考慮人以外的物體的部分。

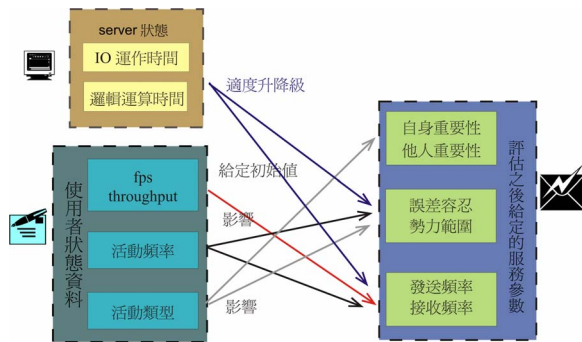
2. **勢力範圍**：勢力範圍則是一個使用者所能感應環境變化的範圍。使用者都會被告知在這個範圍內所有環境中的變動；反之在這個範圍之外的變化，將不會通知這個使用者。

3. **動態路徑誤差容忍**：這個數值代表了使用者所感受到的他人運動的準確度。數值越大代表使用者所認知的他人狀態有可能與事實相距越遠。在他人真實位置與使用者所預測的位置相距超過這個數值才會發送新的資料給使用者。

4. **發送頻率**：系統發送訊息給使用者的頻率，不管有多少訊息累積，若是尚未經過一個發送週期，不管已經累積多少訊息了，還是必須等待下一個時間週期才能發送更新。

5. **接收頻率**：若是 server 端已經十分繁忙了，client 端送再多訊息來也是枉然；因此我們希望能藉由這個參數控制 client 端發送訊息的頻率。

我們以圖三說明使用者狀態的資料與可調整的服務參數之間的關係。FPS 及 Throughput 為使用者客觀的硬體條件限制，因此我們用它們來當作我們所給予服務品質的上限。活動頻率則代表了使用者的專注程度，對於活動較頻繁的使用者，我們假設他會對於「誤差容忍」有較嚴苛的要求，也會期望自己的「勢力範圍」有一定的水準，並且會希望



圖三、使用者滿意度模型與參數調整影響關係圖

系統給予的「發送頻率」和「接收頻率」越快越好，因此這項狀態會影響這幾個服務項目。活動類型的分佈則與使用者關切的偏好相關；動態活動較頻繁的使用者，我們假設其會對於和運動相關的「誤差容忍」及「勢力範圍」有較多的要求。並且他的動態資料對於其他使用者也有比較高的「重要性」。因此，它會影響到「誤差容忍」、「勢力範圍」、「自身重要性」和「他人重要性」這四個服務項目。至於「發送頻率」和「接收頻率」並不會受其影響，因為不論使用者活動偏好為何，我們都該給他合理的服務頻率。

按照上述的關係圖，在系統中我們將這些影響因素量化，利用使用者的狀態計算出期望中最理想的一套服務參數，並且將這套參數值當成「滿意度滿分數值」。換言之，這是一組在我們主觀認定上能讓使用者完全滿意的服務參數。

此套服務參數的量化公式如下：

- 自身的重要性=初始值+(動態活動比-靜態活動比)×調整參數 (1)
- 他人重要性=初始值-(動態活動比-靜態活動比)×調整參數 (2)
- 勢力範圍=初始值+(動態活動比-靜態活動比)×調整參數+(活動頻率-1)×調整參數 (3)
- 動態路徑誤差容忍=初始值-(動態活動比-靜態活動比)×調整參數-(活動頻率-1)×調整參數¹ (4)
- 發送週期=[(throughput+活動週期)/2]×Min(3,1/動態活動比) (5)
- 接收週期=[(throughput+活動週期)/2]² (6)

以上式子都是依據我們之前提出的假設為前提來思考制定的，參數的初始值都可以經由人機介面模組進行動態調整，方便於觀察每一項參數實際上帶來的影響。

接著我們介紹滿意度的計算方法。根據之前所提，我們會針對 server 運作和 client 的狀態來調整

一組服務參數，因此我們擁有兩組參數：《當前服務參數數值》和《滿意度滿分數值》。而我們分別將每一項參數作以下的運算：

對數值越大服務品質越佳的參數而言，
 單項滿意度=Min(《當前服務參數數值》/《滿意度滿分數值》,1) (7)

對數值越小服務品質越佳的參數而言，
 單項滿意度=Min(《滿意度滿分數值》/《當前服務參數數值》,1) (8)

綜合滿意度=Min(所有單項滿意度)/Min(活動頻率+0.5,1) (9)

最後的《綜合滿意度》就是我們對使用者滿意程度的量化數值。其中，我們取單項滿意度的最小值來評估綜合滿意度的理由，是因為我們認為只要有其中一項服務品質過於低落，都會讓使用者無法忍受。而綜合滿意度最後還會再除以一項與活動頻率有關的數值，原因是我們認為活動頻率小於 0.5 之使用者，其對於系統服務品質的敏感度會有下降的趨勢，因此當其活動頻率低於 0.5 時，我們會放大其綜合滿意度的數值。綜合滿意度的數值會在 [0,1] 區間內跳動，我們在此區間內劃出一個《可接受區間》來描述使用者是否能忍受系統的服務品質；例如，我們的實驗中假設《可接受區間》定義為 [0.7, 1]。若是使用者的《綜合滿意度》落在《可接受區間》，我們稱此使用者為《滿意使用者》。

4.2 Client 服務參數初始值與 Server 升降級機制

我們先前提到過，在系統中除了定期根據使用者的差異性來調整服務之外，還會考慮 server 當前的狀況來調整。如圖三所示，整體系統服務參數的給定分成兩個階段：1. 根據 client 狀態給定初始值及 2. 根據 server 狀態調整最終值。我們以下分別介紹這兩個階段運作的方式。

1. 根據 client 狀態給定初始值：初始值的給定基本上是根據前一小節《滿意度滿分數值》來給定。這組數值由於是我們評定滿意度滿分的數值，因此在此把它當成服務品質的上限；不管 server 狀態如何，我們都不會給予比這組數值更佳的服务參數。

2. 根據 server 狀態調整最終值：有了初始值之後，基本上若是 server 的狀態許可，我們會按照這個初始值來服務使用者。但是當 server 資源不足時，我們會開始主動降低所給予使用者的服務品質，等待 server 有充裕的資源，服務品質將會再度調漲。

我們測量負責《運算》和《IO》這兩部分函式執行所需要的時間以及每次執行到下次執行之間的時間間隔來評估 server 目前的狀態。根據目前的狀態，系統會自行將服務品質作適當的調整。

¹此項調整還有個下限是 [0.3]。

²此項參數預設上限 1000 下限 500，原因是假如太過於限制使用者端發送訊息的頻率，將會影響到我們判斷使用者活動頻率的準確度。

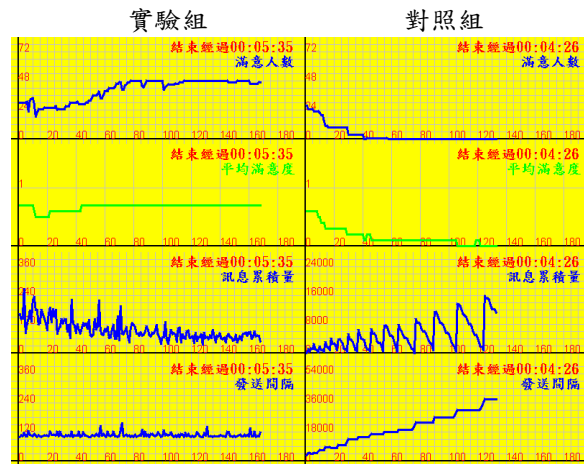
5. 實驗討論與分析

我們在實驗當中搭配了許多不同的軟硬體環境來實驗 IMCTRL 對於整體系統表現的影響，我們以四類統計圖：「滿意人數」、「平均滿意度」、「訊息累積數量」、「系統發送週期」來觀察系統狀態的變化。「滿意人數」指的是整個系統中滿意使用者的數量，而依據前面的定義，滿意使用者是指其綜合滿意度在可接受區間範圍內。而「平均滿意度」則是系統當中所有使用者滿意度的平均值，依據這項數值跟滿意人數可以判斷整個系統中使用者滿意度的標準差。「訊息累積數量」則是目前暫存在系統當中尚未發送出去的訊息數量。而「系統發送週期」指的是系統處理發送訊息的行程其運作一次所耗費的時間，在統計圖中則顯示成「發送間隔」。

圖四是我們所有實驗中具代表性的一個範例，上限人數為 50 名。實驗結果顯示，實驗組的滿意人數呈現上升的趨勢，而對照組則是完全不會有人滿意。從發送間隔來看對照組則呈現無止盡的成長，實驗組則是維持在一定的範圍。整體而言，實驗組表現比對照組好了很多。關於對照組的訊息累積數量線條圖呈現向上攀升波浪狀則是因為我們統計的訊息是確定要發送的訊息數量，並沒有包括使用者將訊息送達 server 時放在等待處理暫存區裡的訊息。

在實驗當中，我們發現不論是處於怎麼樣的網路頻寬下，IMCTRL 都能發揮不錯的效用。但我們也發現，如果使用者本身的行為和硬體能力並沒有差異，則我們的 IMCTRL 就失去了改善效能的功用。不過這是正常並且可以預測到的結果，因為我們改善效能的方式並非增加系統可用的資源，而是更妥善的運用系統的資源，因此當使用者們都需要同樣的服務時，我們自然無法收到截長補短之效果。

另外，我們也從實驗結果得知 server 本身若是有升降級的機制，系統整體的服務也會呈現比較穩定的狀態，這是由於當升降級機制開啟的時候，我們並不是當使用者狀態一更新就立刻將服務品質更動為所評定的數值，而是會配合 server 的狀態慢慢將服務參數的向所評定的數值靠近。而很重要的一點是，在 server 資源嚴重缺乏的時候，即使以使用者的角度看起來都是處於不滿意的狀態。但是有升降級機制的 server 代表了整個網路服務是在我們的掌握之中。我們並非任憑資源耗竭來反應服務的低落，而是藉由得知系統資源的不足主動將服務品質調降。有掌握服務品質的能力代表著我們可以調整我們的資源分配方式。在本篇論文當中我們是依照使用者的滿意度來調整，但當然也可以轉換成依照其他考量來調整網路服務。從實驗當中也可以發現，三項主要的服項目：「重要性」、「動態路徑預測」和「勢力範圍」也都各有其效用，而其發生效用的場合也都跟我們預期的吻合。重要性在使用者有些好動有些文靜的情況下有其效用；動態路徑預測則是只要使用者有運動就能發揮功用；而勢力範圍的



圖五、實驗統計圖範例

篩選在使用者分散於不同區域時就能發揮不錯的效果。後兩者已經是很普遍的概念，雖然實作方式有很多種，但是過去都是以同樣的標準來服務每個使用者，每個使用者都是一樣的勢力範圍、一樣的預測誤差容忍值。但在我們的實驗中證實，在某些情況之下，依據使用者的狀態來調整這兩值，的確可以改善 server 的效能。

另一項也很重要的調整是「發送週期」，從實驗當中我們可以很清楚的知道，即使沒有依據使用者活動的方式作任何篩選機制的調整，只針對使用者的硬體能力來改變發送訊息的週期，在使用者硬體能力差異度極大的時候，依然有它明顯的功效。

我們也觀察到「訊息覆蓋」這項機制的重要性。在使用 IMCTRL 的 server 和最原始的 server 實驗比較之中，我們可以發現原始 server 的訊息累積數量不斷的向上攀升；雖然在 server 資源不足的時候使用者同樣都是不滿意，但是訊息不斷累積所代表的含意則是使用者所收到的訊息將是很久之前的過去所發生的事件訊息。雖然在我們的滿意度模型當中並沒有納入訊息延遲的考量，但是顯而易見的訊息累積是個不良的現象，因此同類型的訊息覆蓋機制有其存在的必要。

綜言之，實驗結果讓我們得知在我們滿意度的定義之下，我們實作的系統的確能讓使用者更滿意，並且節省系統的資源；而各項差別服務項目，也都有其功用。撇開滿意度模型的客觀性問題不談，實驗結果也證實了我們的系統確有控制網路服務品質的能力。

6. 目前問題與未來工作

6.1 目前問題

首先要提到的是滿意度模型的問題。目前滿意度模型當中各項狀態的對應影響關係，都是由我們主觀假設提出的。雖然這些想法直觀合理，但是有些地方可能不夠周延完備；例如前一小節所提出的訊息延遲問題。這個問題很明顯會影響到使用者的

滿意程度，但是在我們的模型當中並沒有將它納入考慮，因此滿意度的模型仍有可以改進的空間。另外，除了模型本身的完整性，量化數值的調整也是另一個議題，在本篇論文當中並沒有反覆測試並提出最佳數值的建議。在評估滿意度的公式方面，我們計算的方式是採用滿意度滿分數值和當前服務參數兩者作單純的除法，但是使用者的感受是否是如此單純的線性關係這個問題也有待商榷。

關於我們所實作的重要性篩選，其使用方式也是一項需要進一步討論問題，因為雖然重要性的篩選能很有效的篩選訊息，但相對的如何認定何謂「重要」就是一個困難的議題。以系統開發者主觀的角度來定義模型已經不容易，若是希望把使用者的感受包含進來考量更是一項艱鉅的挑戰。在我們系統當中，所有使用者都是使用同樣的基底，再根據他們的狀態作服務的調整，或許可以考慮讓使用者表達自己對於服務品質的要求來作我們調整服務的參考，但這在整個虛擬環境的互動類型有明確定義之下比較有其意義，在互動類型尚不明確的情況下，使用者也無從考量他需要怎樣的服務。

在動態路徑預測上面，我們使用簡單的演算法，因此可能預測效果不良而導致實驗結果比使用良好演算法來的差。在運算勢力範圍方面也是一樣，都是有實作其功能，卻沒有加以最佳化。在 server 人數大量提升的時候會不會導致運算的負擔太重也需要進一步實驗。以目前的狀況看來都是網路資源比 CPU 資源先發生耗竭的現象，也因此我們在本篇論文沒有考慮將運算的部分作些最佳化的處理。

另外跟延遲問題相關的一個問題就是同步化，目前的系統完全沒有考慮到同步化的問題。而在多人虛擬環境系統之下，有許多互動類型是需要非常精確的訊息同步機制的。如需要同步機制，則我們目前的篩選機制勢必要跟著作些調整。

不過這些問題我們相信在將來都是可以解決的，但許多細節並不是本篇論文專注的焦點，因此我們在此並沒有加以處理。

6.2 未來工作

在未來我們希望能夠針對使用者的感受作更精確的分析，例如實際統計使用線上活動的方式和他感受之間的關連性。在評估滿意度的方式上，我們也希望透過夠廣泛的真人實驗和統計，找出比較客觀的滿意度評估方式，藉此提升我們系統控制服務的正确性。

我們也希望能夠將這套系統作更廣泛的延伸和應用，例如嘗試與多 server 的架構結合，擴展系統的可用性，並且透過與底層 IMNET 和其動畫元件的整合以及搭配 IMAGENT[]當中情緒自主式的虛擬演員，能夠實作一個有明確互動方式和訴求的多人虛擬環境系統。

在 IMCTRL 本身的調整機制上面，我們也期望能用更動態的方式讓系統建構者來改變規則。例如

評估的公式、滿意度模型的關連建立和系統內部訊息的分類等內部的機制，若是能用文字檔之類的外部描述來更改，則可以讓系統因應不同類型的虛擬環境系統作不同的服務調整方式。

另外關於同步化方面的問題，我們也期望能在 IMCTRL 上面整合出一套可行的解決方案。在現行的架構下，由於訊息覆蓋機制的緣故，許多訊息是會被覆蓋的，也有許多訊息是會延遲發出的，該如何區分需要同步的訊息也是另一個未來可以延伸的工作。

最後，在一些實作的技術上，例如動態路徑預測，我們也可以採用更好的演算法，而勢力範圍的運算方式，也已經有許多不同的空間切割方式可以減少運算所耗費的時間，這些都是可以在未來加以改進的部分。

參考文獻

- [1] B. Blau, C. E. Hughes, M. J. Moshell, and C. Lisle, "Networked Virtual Environments," *ACMSIG-GRAPH Special Issue on 1992 Sym. on Interactive 3D Graphics*, Cambridge, MA, pp.157-164, 1992.
- [2] W. Cai, P. Xavier S. J. Turner, and B.-S. Lee, "A Scalable Architecture for Supporting Interactive Games on the Internet," *Proc. of the sixteenth workshop on Parallel and distributed simulation*, pp.60-67, 2002.
- [3] C. Carlsson and O. Hafsand, "DIVE - a Multi-User Virtual Reality System," *Proc. of the IEEE Virtual Reality Annual Intl. Symp.*, pp.394-401, September 1993.
- [4] C. Faisstauer, D. Schmalstieg, W. Purgathofer, "Scheduling for Very Large Virtual Environment and Networked Games Using Visibility and Priorities," *Proc. of Fourth IEEE Intl. Workshop on Distributed Simulation and Real-Time Applications (DS-RT'00)*, pp.25-27, August 2000.
- [5] S. Fiedler, M. Wallner and M. Weber, "A Communication Architecture for Massive Multiplayer Games," *NetGames2002*, Braunschweig, Germany, April 16-17 2002.
- [6] T. A. Funkhouser, "RING: A Client-Server System for Multi-User Virtual Environment," *ACM SIG-GRAPH Special Issue on 1995 Symp. on Interactive 3D Graphics*, Monterey, CA, pp.85-92, 1995.
- [7] M. Grabner, "WebCAME: A light-weight modular client/server multiresolution rendering system," *Proc. of the eighth Intl. Conf. on 3D Web technology*, pp.63-72, 2003.
- [8] C. Greenhalgh and S. Benford, "Awareness and Interaction in Collaborative Virtual Environments," *Proc. of IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, Cambridge Massachusetts, pp.193-198, June 1997.
- [9] S. Han, M. Lim and D. Lee, "Scalable Interest

- Management Using Interest Group based Filtering for Large Networked Virtual Environments,” *Proc. of ACM Symp. on Virtual Reality Software and Technology*, Korea, pp.103-108, October 2000.
- [10] D. Lee, M. Lim, S. Han, “ATLAS-A Scalable Network Framework for Distributed Virtual Environments,” *Proc. of the 4th Intl. Conf. on Collaborative Virtual Environments*, Bonn, Germany, Sept. 30-October 2, 2002.
- [11] M. Macedonia, M. Zyda, D. Pratt, D. Brutzman and P. Barham, “Exploiting Reality with Multicast Groups,” *IEEE Computer Graphics and Application*, pp.38-45, September 1995.
- [12] L. P. Pantronics and L. C. Wolf, “On the Suitability of Dead Reckoning Schemes for Games,” *NetGames2002*, pp.79-84, April 2002.
- [13] S. Singhal and M. Zyda, *Networked Virtual Environments Design and Implementation*, Addison Wesley, July 1999.
- [14] W.-T. Wong, Z.-R. C., Y.-C. Chang, “A Hybrid Scheme for Massive Multiplayer Online Games using Mobile Agent on Dynamic Quadtree Architecture,” *Proc. of National Computer Symp.*, Taiwan, 2003.
- [15] 傅光聖, An Enhancement of DIS Protocol for Entity State in Networked Virtual Environment, 台灣大學資訊工程研究所碩士論文, 2000.
- [16] IMNET,
URL:<<http://imlab.cs.nccu.edu.tw/IMNet>>.