

第六章

信賴度模擬實驗

我們期望能定義出一個具代表性的測試案例 (Test case) 作為信賴度實驗模擬的基準，並且利用第五章所設計的動態模擬器實做本文所提出的新信賴模型與 D-S 證據推論信賴模型。這一章主要分為四個部分，第一部分運用簡單的例子，配合社會學經常用以評量人際關係的指標「疏離感」(Sense of distance) [8]，先行簡單驗證動態模擬器「確實在模擬的過程中，提供了一條節點間信賴度量化後的觀察途徑」，也就是動態模擬器觀察節點之間關係變化的能力；第二部分我們設計一個供正式模擬用的一般性測試案例，並介紹它所描述的情境，包含節點集合描述、服務集合描述以及計算任務如何判定為「成功」(Success)、「失敗」(Failure) 或是遭受「詐欺」(Deception)；第三部份描述並解釋「作為效益評比標準的公共生產力面向」以及「作為人際關係評比標準的社會學指標」(降低不確定性與社會滲透論) 在模擬中所代表的實際涵意，我們期望能利用這些評比的面向來驗證新信賴模型與現有信賴模型的效能；最後一個部份則是顯示模擬執行之後的結果，嘗試去解釋模擬結果呈現的數據，並藉以驗證本文所提出的新信賴模型是否達到改善現有信賴機制的期望。

6.1. 簡單驗證動態模擬器觀察節點之間關係變化的能力

本文所引用的「社會學行動理論」中對於群體中人與人之間的親疏遠近關係，常常用「疏離感」(Sense of distance) [8] 一詞來形容。依照這樣的社會學

定義，我們觀察人類社會行為可以發現，當群體中的兩個個人之間的好感或信賴度增加之後，兩個個人之間的距離感確隨之遞減。將這樣的理論基礎落實在本文所設計的動態模擬器上，使得本文所設計的動態模擬器擁有其他模擬器所沒有的能力，也就是可以即時動態觀察兩兩節點之間的信賴度高低變化。我們設計了兩個簡單的例子來做說明，分別是僅存在一個提供優良服務節點的情境，與存在三個提供優良服務節點的情境，兩個例子都簡單地包含 20 個以上的獨立節點與一個以上的服務。展示的過程中搭配本文所提出的新信賴模型來做模擬，目的在於能夠順便驗證新信賴模型具有消弭熱點的能力。

在第一個例子中，我們嘗試在小世界中置入一個提供良好服務 S 的節點 P_1 ，並使其他節點隨機且重複發生需求服務 S 。如下圖 6.1 連續的節點運動軌跡圖所示，依序由左而右由上而下，我們可以看到除了 P_1 之外的其他節點，因為請求節點 P_1 的服務並給予正面評價，使得所有節點漸漸地提升對節點 P_1 提供服務 S 品質的信賴度，也漸漸向 P_1 靠攏聚集，形成社會學上的「群聚現象」(Industrial clustering) [9]。首先，我們預期雖然剛開始的時候僅有少數幾個節點知道節點 P_1 的存在並提出服務要求，並且因為使用度與評價還沒有很高的緣故，使得我們從圖中觀察到很多節點依舊距離節點 P_1 很遠。

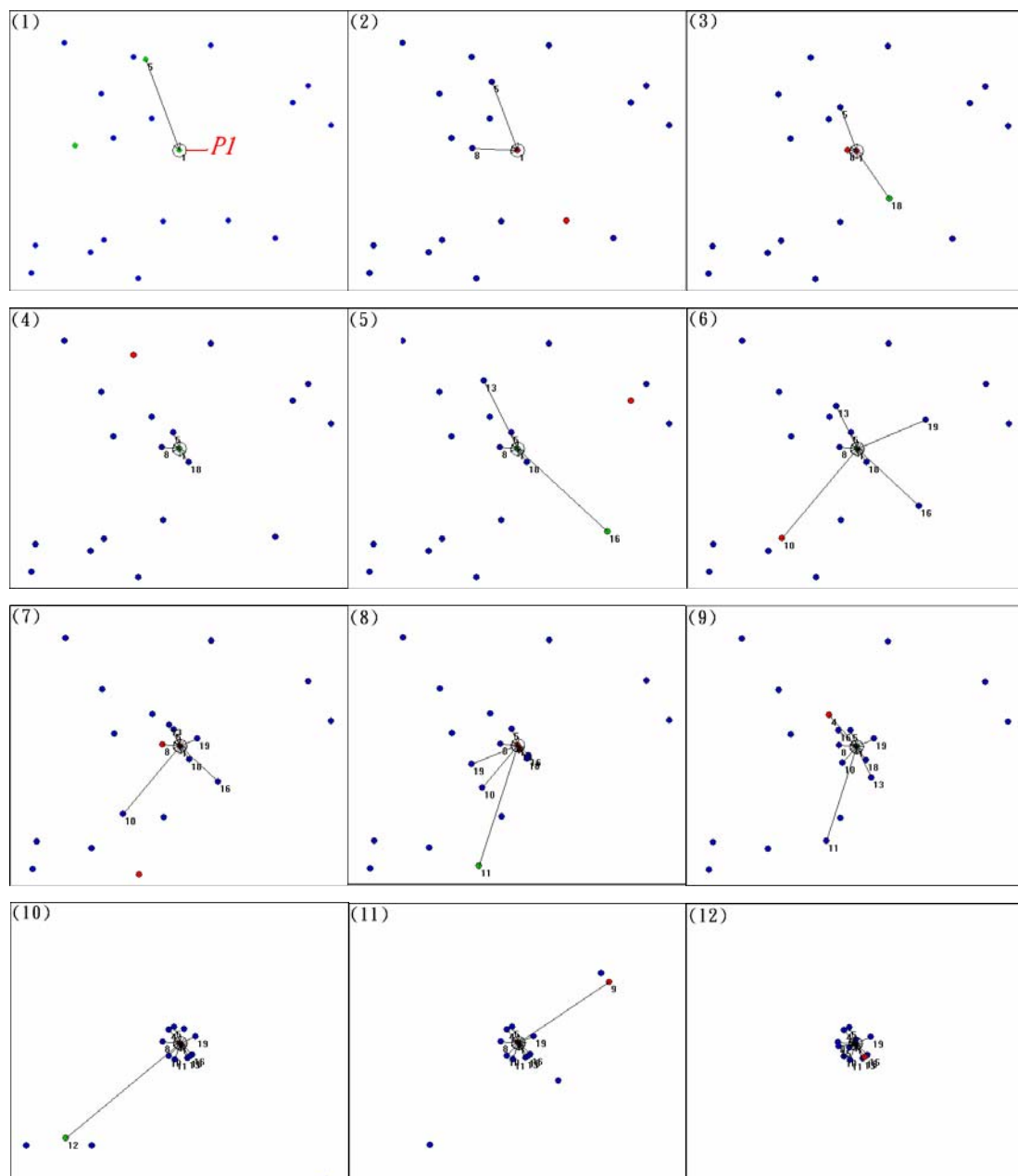


圖 6.1，動態模擬器模擬存在單一優良節點的動態軌跡連環圖。

如編號 (1) 所示，我們將想要觀察的優良節點 P_1 移至觀察區域中心位置。編號 (1) 至編號 (3) 顯示了整體小世界在模擬的初期，因為缺乏熟識的節點可供推薦機制運用，僅僅極為少數的節點藉由主動地向外認識新朋友得知節點 P_1 的存在，並交付計算任務給 P_1 。

隨時間經過，因為使用的次數遞增，使得節點 P_I 口碑上升，漸漸地讓其他的節點也因為相信口碑開始向節點 P_I 提出服務要求。我們可以從上圖 6.1 中的編號 (4) 到編號 (9) 觀察到，因為整體小世界對節點 P_I 所提供的服務 S 正面評價不斷增加，原本就使用節點 P_I 所提供的服務 S 的其他節點，因為持續使用且服務需求都被滿足，使得對節點 P_I 的信賴度不斷累增，與節點 P_I 之間的距離也越來越短；其他新使用 P_I 服務的節點也因為整體小世界給予的高評價產生高度信心。最終所有使用節點 P_I 服務的節點都聚集在 P_I 四周圍，逐漸地往 P_I 靠攏。最後整體小世界對節點 P_I 所提供的服務 S 正面評價不斷增加，原本就使用節點 P_I 所提供的服務 S 的其他節點，因為持續使用且服務需求都被滿足，使得對節點 P_I 的信賴度不斷累增，與節點 P_I 之間的距離也越來越短；其他新使用 P_I 服務的節點也因為整體小世界給予的高評價產生高度信心。最終所有使用節點 P_I 服務的節點都聚集在 P_I 四周圍，形成預期中的群聚關係。圖 6.1 中編號 (10) 到編號 (12) 顯示了此一良性循環加速作用的過程，我們可以觀察到比編號 (7) 到編號 (9) 更快速的群聚現象。（由實驗數據得知，編號 (7) 到編號 (9) 平均花費約莫 15 秒，而編號 (10) 到編號 (12) 平均花費僅 4.7 秒。）

這裡我們只看到「因為信賴度的增加而向提供優良服務節點靠攏」的現象，事實上即使是提供優良服務，節點本身依舊存在著負載與離線等等無法完成計算任務的因素，使得有服務需求的節點也會出現「因為信賴度衰減漸漸離開原先靠攏的優良節點或是靠向另一個提供優良服務的節點」的現象。關於這一點，我們將以第二個例子來做說明。在第二個例子中，我們設計了同時存在三個提供優良服務的節點的狀態，這使得有服務需求的節點會在這三者之間做出選擇而出現「靠攏」、「遠離或改靠向另一節點」的現象。

再者，我們將節點 P_I 的朋友數目、服務的節點數目以及整體小世界對節點

P_1 的平均評價做成下圖 6.2，嘗試解釋上述所展示的動態模擬器拓樸圖形節點之間信賴度與距離的變化。

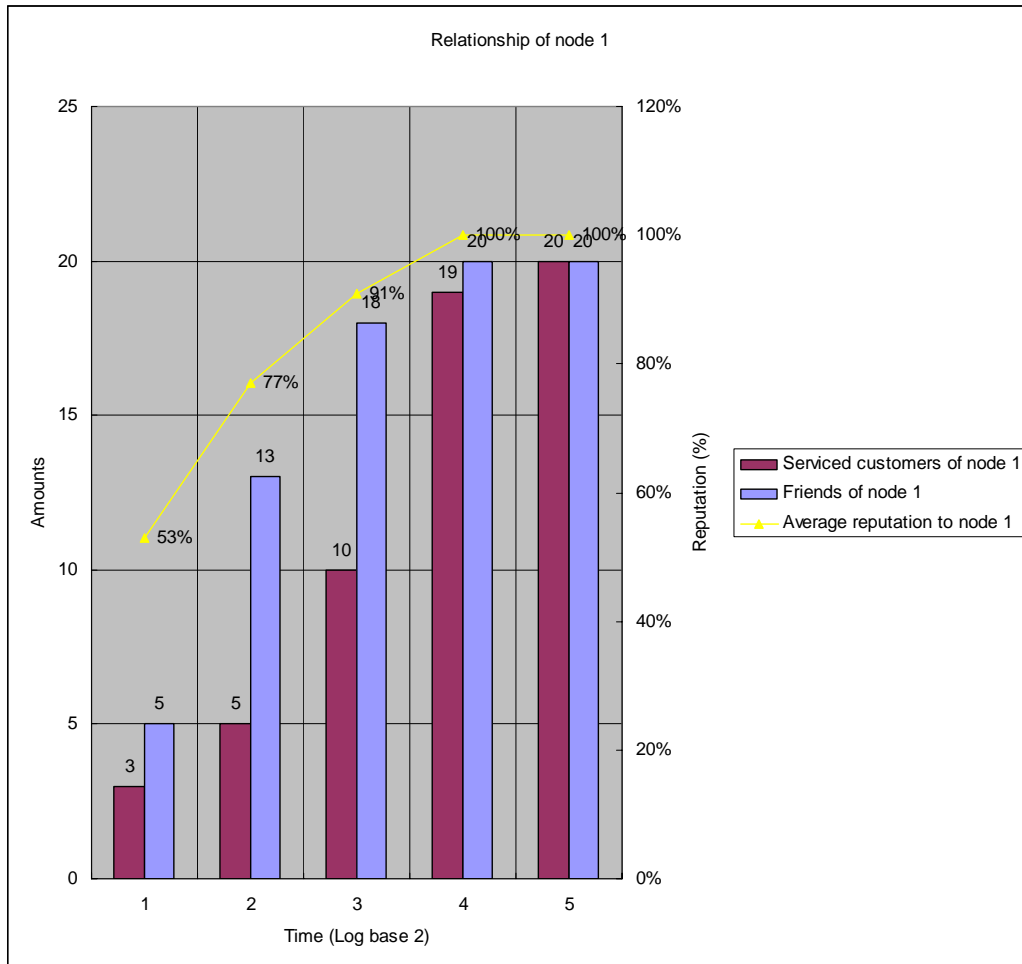


圖 6.2，顯示節點 P_1 的朋友數目、服務過節點數目以及小世界給予的評價。

可以從圖 6.2 中觀察得知，隨著時間經過，因為使用節點 P_1 的服務並給予正面評價的「熟識」（即與節點 P_1 成為好友）節點數目增加，使得「耳聞口碑而向 P_1 請求服務的節點」數量增加，並持續給予正面評價，形成一個良性循環。整體小世界對節點 P_1 的正面評價愈高，單一節點對節點 P_1 的信賴度就愈高，與節點 P_1 之間的距離也就愈短；相較於社會學的觀點，我們可以說單一節點對於節點 P_1 的信賴度提升，縮短了對節點 P_1 的「距離感」。此一觀察而得的現象可說是非常符合我們對「引用社會學行動理論而實做出來的動

態模擬器」的期望。

在第二個例子裡，我們嘗試將提供優良服務的節點數目增加到三個，分別是節點 P_1 ，節點 P_{10} 與節點 P_{20} ，同樣以社會學理論所談到的群聚現象為觀察重點，如圖 6.3 中連續節點的運動軌跡圖所示，依序由左而右由上而下。不同的是，在這個例子中，我們還期望可以從動態模擬器觀察到當節點 P_1 所能服務的節點數目飽和時，是不是能夠順利建立起本文所提出的新信賴模型描述的「轉手路徑」，進而將無法負荷的計算任務轉介給另外兩個提供同樣優質的服務節點，讓整個小世界趨於一個平穩狀態的現象。首先，我們預期在這個例子中可以觀察到一個與「僅僅存在單一優質服務節點」時的不同之處，那就是原先因為口碑推薦而將計算任務交付給節點 P_1 的其他節點，在 P_1 呈現高負載（節點呈現紅色）的狀態並了解自己無法負荷此計算任務後，會因為新信賴模型「同業公會機制」發生作用，快速地找到同樣提供優質服務的其他節點，並將計算任務轉介給它們。我們設定三個提供優良服務的節點並置入模擬的小世界中： P_1 、 P_{10} 與 P_{20} 。

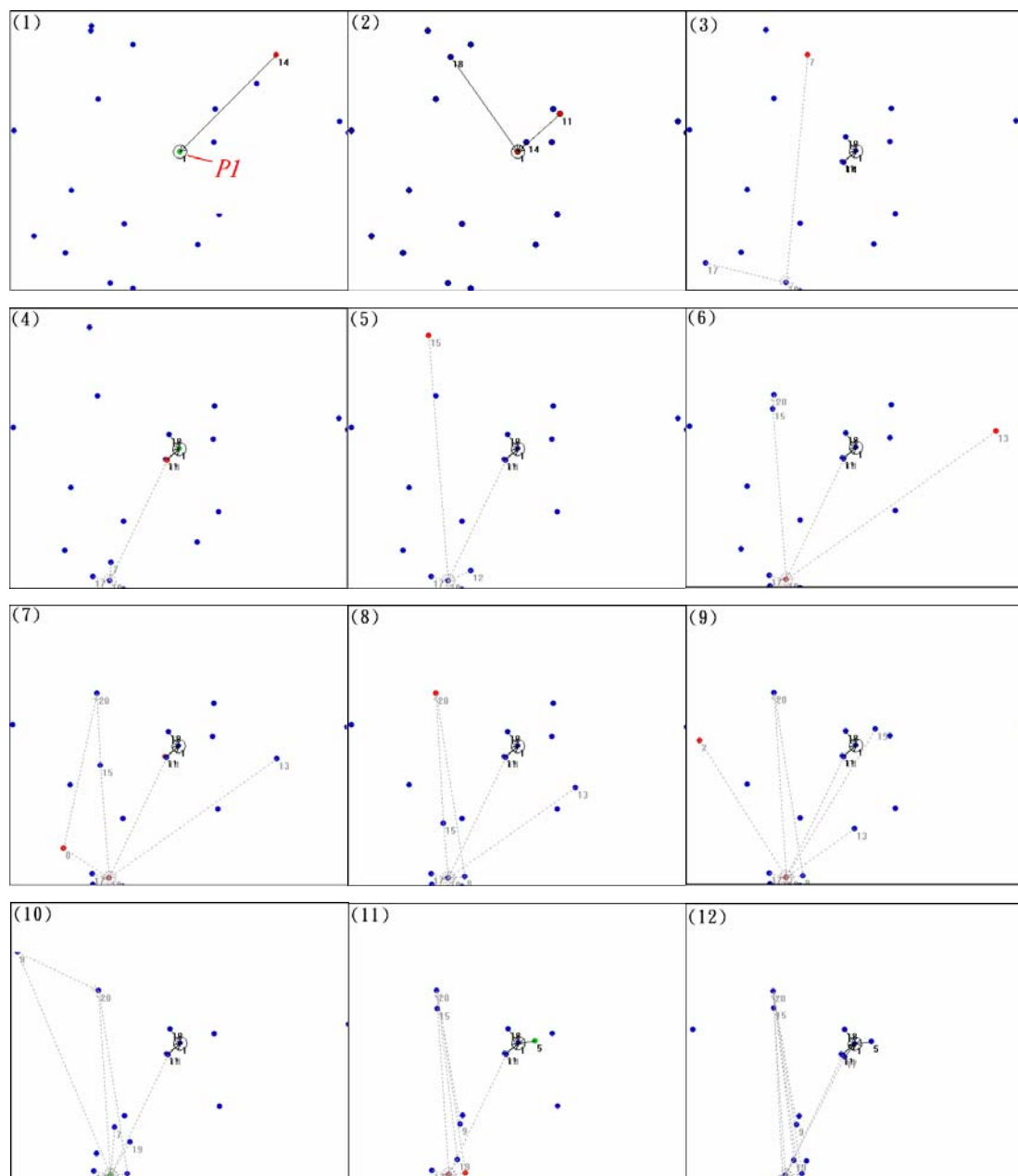


圖 6.3，動態模擬器模擬存在三個優良節點的動態軌跡連續圖。

如編號 (1) 所示，我們將想要觀察的優良節點 P_1 移至觀察區域中心位置。編號 (1) 至編號 (2) 顯示了整體小世界在模擬的初期，因為缺乏熟識的節點可供推薦機制運用，僅僅極為少數的節點藉由主動地向外認識新朋友得知節點 P_1 的存在，並交付計算任務給 P_1 。對照僅存在一個優良節點的例子，我們發現在編號 (3) 時便已經出現同業公會的影響，使得小世界中有服務需求的節點不再只向 P_1 提出要求，而會向節點 P_1 之外的優良節點尋求協助。

隨著時間經過，我們發現由於通過新信賴模型的「同業公會」機制，「轉手路徑」可以很快地被建立起來，於是我們透過動態模擬器可以很輕易地觀察到節點 P_1 雖然得到小世界裡最高的評價，但是卻沒有維持在高負載的情況，而是與節點 P_{10} 、 P_{20} 互相輪流，同時維持計算任務順利運行與負載量，在短時間之內消弭了「熱點」（服務瓶頸）的現象。編號 (4) 到編號 (9) 記錄了有服務需求的其他節點在三個提供優良服務的節點 P_1 、 P_{10} 與 P_{20} 間，因為所得到的信賴度評價不同而做出不同選擇的現象。於是，三個提供優良服務的節點 P_1 、 P_{10} 以及 P_{20} 在一個「完成計算任務、得到正面評價、得到更多的計算任務並完成」的良性循環之下，不斷提昇在小世界中的評價。我們以節點 P_1 為觀察節點，以節點 P_1 與其他節點（ P_{10} 與 P_{20} 除外的節點）的信賴度為半徑，我們可以得到如圖 6.3 顯示的拓樸圖形。如同在觀察「僅存在單一優質服務節點」時一樣，編號 (10) 到編號 (12) 顯示了群聚現象快速地在節點 P_1 、 P_{10} 與 P_{20} 四周出現。群聚現象之外，節點 P_1 、 P_{10} 與 P_{20} 輪流降低負載量的現象，也相當符合我們對新信賴模型與動態模擬器的預期。

我們一樣將上述的觀察過程中，節點 P_1 、 P_{10} 以及 P_{20} 的朋友數目、服務的節點數目與小世界給予的平均評價作成圖 6.4，顯示如下。

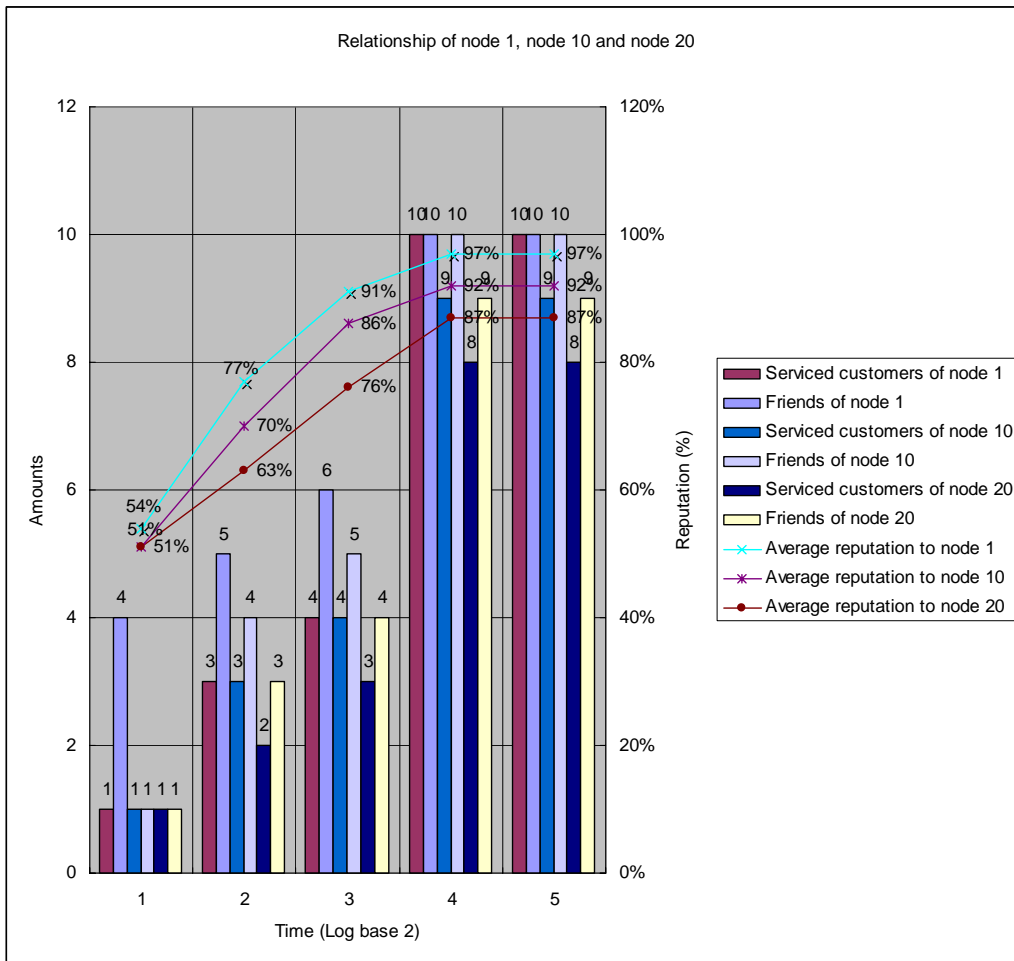


圖 6.4，顯示節點 P_1 ， P_{10} 與 P_{20} 三者的朋友數目、服務過節點數目以及小世界給予的平均評價。

我們發現除了一開始就被詢問的節點 P_1 之外，雖然 P_{10} 與 P_{20} 都有被詢問並要求提供服務，但最後趨於穩態時節點 P_{10} 與 P_{20} 的信賴度並沒有完全地百分之百，且認識的朋友數目與服務的節點數目也不盡相同。這顯示了有些節點最終會選擇 P_1 、 P_{10} 或是 P_{20} 三者之一作為提供服務節點，而不會只向評價最好的服務節點提出服務要求。也驗證了新信賴模型對於消弭熱點發生的能力。

6.2. 測試案例基本定義

為了忠實模擬現實世界中分散式網路運算架構，首先必須清楚地定義何謂

「完成一個計算任務」。我們定義一個運算任務的「完成」必須是歷經一個「交付 (Hand over)」與「應答 (Response)」的過程，透過請求網路上某一個「任務承接者 (Task receiver)」的服務，將計算任務交付給任務承接者並經過計算，最後再回到原「任務擁有者 (Task owner)」手上，由任務擁有者來判定此計算任務為「成功 (Success)」、「失敗 (Failure)」或者是遭受任務承接者的「服務詐欺 (Deception)」。

這裡所標定的任務擁有者與任務承接者的角色並非絕對，僅僅是說明在某一個時間點上，服務要求者 (任務擁有者) 與服務提供者 (任務承接者) 在計算任務傳遞上的相對關係。圖 6.5 顯示模擬器模擬一個任務從對任務承接者發出服務請求到執行結束回到任務擁有者手上的簡易時序圖 (Sequence diagram)。



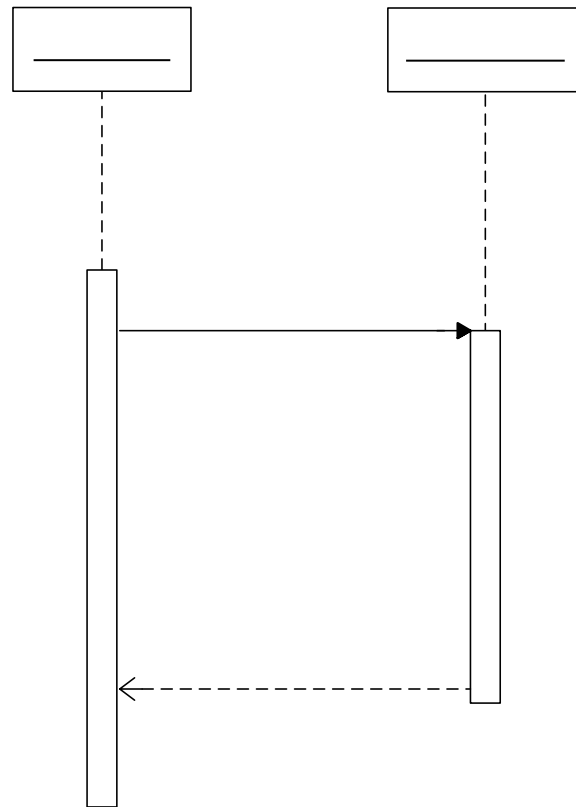


圖 6.5，模擬器模擬一個任務從發出服務請求到執行結束回到任務擁有者手上的簡易時序圖（Sequence diagram）。

我們簡化現實世界中「詐欺」的定義，讓整個測試案例的模擬過程可以將焦點集中在因不同的信賴模型對整體小世界產生的不同影響上。我們定義當一個計算任務經由任務承接者提供計算服務後再度回到任務擁有者的身上時，此時任務擁有者根據「任務交付時，任務承接者所提供的任務計算時間契約（Promised executed time）是否大於或等於最後計算任務實際所花費的計算時間」來做為任務擁有者是否受到任務承接者詐欺的依據。「計算時間契約」表示任務承接者給予任務擁有者的承諾，與任務承接者的實際負載（Loading）或是工作佇列是否滿溢（Full）等客觀環境條件無關。

除此之外，我們亦簡化針對計算任務結果為「成功」或是「失敗」的判定。我們定義一個被任務擁有者 A 所擁有的計算任務 X 必須同時符合下述條件才能被判定為任務成功：

- 1) 任務擁有者 A 必須透過信賴模型所定義的推薦制度推薦後，得到一個服務品質在可接受範圍內的服務提供者 B 提供服務。
- 2) 服務提供者 B 的工作佇列必須處於非滿溢狀態，以便容納計算任務 X 進入排隊等待，否則 A 必須另外尋找服務提供者。
- 3) 當服務提供者 B 符合上述兩個條件後，任務擁有者 A 便會將 B 視為任務承接者，並將計算任務 X 進入交付給 B 。交付完成後，計算任務 X 必須在 B 的工作佇列中排隊等待 B 從佇列中取出執行。
- 4) 當任務承接者 B 取出 X 並執行後，便將 X 回應給原任務擁有者 A 。倘若此時經 A 判斷計算任務未受到服務詐欺時，此時便可判定計算任務 X 為「成功」。

同樣地我們也針對計算任務的失敗做了定義。我們定義一個被任務擁有者 A 所擁有的計算任務 X 必須同時符合下述條件時才能被判定為任務失敗：

- 1) 任務擁有者 A 必須透過信賴模型所定義的推薦制度推薦後，卻遲遲無法得到一個服務品質在可接受範圍內的服務提供者 B 提供服務。
- 2) 計算任務透過信賴模型的推薦機制徵詢過所有可詢問的節點後，若返回原任務擁有者 A 的手上時仍沒有節點可以提供服務，此時便判斷計算任務 X 為「失敗」。

「任務擁有者」與「任務承接者」均是一包含於構成小世界的節點集合 Θ 中的獨立節點。在本測試案例中，節點集合 Θ 的定義如下所示：

$$\Theta = \{ P_1, P_2, P_3, \dots, P_i \}$$

其中， i 為任意非零正整數。

在我們對測試案例所做的基本定義中，節點集合 Θ 必須包含一個以上（含一個）的獨立節點。在模擬開始之初每一個集合中的節點相互之間並不認識。為了讓動態模擬器針對不同的信賴模型所得到的模擬結果，不會因為模擬的任何一個信賴模型受到節點在先天條件設計不良，進而造成最後觀察有所謬誤，我們定義集合 Θ 中的任一節點均擁有一個以上的非詐欺服務，且工作佇列長度必須大於或等於 2，以便容納一個新任務以及一個「正好回到自身節點的舊任務」同時存在。這裡所謂的「正好回到自身節點的舊任務」指的便是前面描述「任務擁有者 A 透過推薦機制，將計算任務 X 交付給任務承接者 B。當 B 完成計算後，將 X 返回給原任務擁有者 A」的「交付」與「應答」結果。由於最終計算任務必須回到原任務擁有者的身上，以便判斷任務是否為「成功」、「失敗」或是「詐欺」，所以每一個節點都必須至少保持一個工作佇列空位來容納這個等待接受評判的舊任務。

我們定義整個小世界在模擬的過程中所有出現的服務需求共有 j 個不同的服務，均包含在一個服務集合 \mathcal{E} 之中，其定義描述如下：

$$\mathcal{E} = \{ S_1, S_2, S_3, \dots, S_j \}$$

其中， j 為任意非零正整數。

我們隨機給定每一個節點所擁有的服務內容與個數，並確保 \mathcal{E} 中所定義的每一項服務均會被 Θ 中一個或一個以上的節點所擁有。在這樣的模擬情境下，當模擬器從外部不斷接收我們事先描述好的「觸發事件」（指定任一個定義於 Θ 中的節點 P_i 提出任一包含於 \mathcal{E} 中的服務需求 S_j ）時，不論目前觀察的信賴模型為何，我們可以保證所觸發的服務需求都能得到至少一個以上的節點提

供服務，而不會因為缺少某項服務提供者使得觀察有所偏頗。

我們定義動態模擬器所接受的「觸發事件」的事件集合 ρ 如下所示：

$$\rho = \{ E_1, E_2, E_3, \dots, E_k \}$$

其中， k 為任意非零正整數。

為了滿足模擬的需求，任一個包含於事件集合 ρ 中的單一事件 E_k 均必須由定義好的格式組合而成：「觸發事件編號 (EID)」、「執行動作種類編號 (ACT)」、「要求服務種類編號 (ACTRS)」、「執行時間承諾 (NEEDED_PROMISED_WDS)」以及「任務擁有者編號 (NNO)」。各項定義描述如下：

- 1) EID：紀錄觸發事件的編號，以便使用者透過動態模擬器觀察已經觸發或是正在觸發的事件內容。基本描述格式為「EID:事件編號」。
- 2) ACT：觸發事件所代表的「動作 (Action)」。我們事先定義了幾項基本動作，並將動作賦予編號。以一般性的測試案例為例，我們希望可以透過動態模擬器觀察到「小世界中節點之間因為提出服務要求而造成的信賴度變化」，所以輸入動態模擬器中的觸發事件便會以「要求某項服務」這樣的動作為主，將其簡化為我們定義的格式，便成為「ACT:4」的字串。其中，「4」代表我們賦與「要求某項服務」這個動作的編號。基本描述格式為「ACT:動作編號」。
- 3) ACTRS：紀錄觸發事件將要執行的動作為「要求某項服務」時的服務編號。搭配第二點所描述的「ACT」而成的格式。也就是說，當某一觸發事件描述裡面出現「ACT:4」的時候，表示此一觸發事件所觸發的動作為「要求某項服務」，此時搭配使用「ACTRS:1」即表示，所要求的服務為編號 1 號的服務；換言之，若「ACT」所觸發的事件並不是「要求某項服務」

時，「ACTRS」便毫無意義。基本描述格式為「ACTRS:服務編號」。

4) **NEEDED_PROMISED_WDS**：紀錄觸發事件「至多」只能接受多少的執行工作時間承諾。當任務擁有者某一項執行任務必須尋求其他節點的協助完成時，任務擁有者便會向同樣在小世界中的其他好友詢問「是否有節點可以承諾以小於 **NEEDED_PROMISED_WDS** 的執行工作時間協助完成任務」。當任務擁有者因為推薦得到某一個服務提供者願意成為任務承接者，並提供服務協助解決計算任務時，此時任務擁有者便會與任務承接者依照 **NEEDED_PROMISED_WDS** 所記錄的執行工作時間承諾訂下「契約」，作為將來任務承接者完成任務並將任務返回任務擁有者時，任務擁有者用以判斷是否受到任務承接者「詐欺」的依據。基本描述格式為「**NEEDED_PROMISED_WDS**:執行工作時間承諾」。

5) **NNO**：此一觸發事件所指定之任務擁有者的節點編號。「**NNO**」可以為節點編號，或是紀錄「0」表示隨機指定節點作為任務擁有者。以一般性的測試案例而言，我們為了忠實模擬分散式計算網路的真實性，以及測試接受模擬的信賴模型的「強壯性 (Robust)」，我們傾向使用隨機指定給某一節點作為任務擁有者。基本描述格式為「**NNO**:指定的節點編號」。

綜合上述五項格式，舉例而言，當我們定義一個觸發事件 E_k ：

$$E_k = EID:1;NNO:0;ACT:4;ACTRS:1;NEEDED_PROMISED_WDS:2$$

則此觸發事件具有下列幾項涵義：

- 1) 觸發事件編號為 1。
- 2) 此觸發事件將隨機指定給目前節點集中的任意一個節點。
- 3) 此觸發事件將令任務擁有者提出服務需求：要求擁有服務編號為 1 的優質節點提供服務，並成為任務承接者。
- 4) 此觸發事件將要求任務承接者必須遵守執行工作時間為 2 的承諾。

最後，便是我們將測試案例中的每一個節點所能「記憶住」(Remember)的好友個數以及使用過的名單限制在一個合理的範圍之內。我們不希望每一個節點記住所有的「曾經是好友，但是很久沒有再接觸過」的節點，或是「曾經使用過，但是很久沒有再去使用」的服務。主要的理由是如果每一個節點都可以擁有記住所有曾經交往的朋友或使用的服務，這不但不符合我們現實生活中的經驗（長時間沒有聯絡的朋友或服務，便會失去聯絡或遺忘），長時間下來使用信賴模型所建置的推薦機制便會失去意義。

我們定義節點集合 θ 中的每一個節點所能紀錄的好友個數如下所示：

$$\eta = \{ F_1, F_2, F_3, \dots, F_l \}$$

其中， l 為任意非零正整數。

之後我們依照上述的定義設計出一個完整的測試範例，供動態模擬器針對本文所提出的新信賴模型以及 D-S Evidence Theory 信賴模型做即時模擬的依據標準。

6.3. 公共生產力評比與社會學人際關係衡量指標

針對模擬的實驗結果，一方面我們利用第一章所提到的「公共生產力」[2][12] 來做為衡量測試案例在相同的輸入及環境條件下，因為使用不同的信賴模型所產生的輸出效益；另一方面採用第三章所提到的「降低不確定性理論」與「社會滲透論」兩項在社會學上用以衡量人際關係發展的指標性理論，來驗證

新信賴模型是否因為引用了社會學行動理論而產生不同的輸出效益。本文在稍後的比較裡所用的「分數」都是相對的概念，只是為了將節點之間的信賴度變化加以度量化，以便可以清楚地看出兩個不同的信賴模型之間的差異性。

我們將公共生產力原先的四個面向：目標達成度、社會公平、服務品質，以及服務滿意度加以細分並實做，我們可以得到五種衡量標準：「計算任務成功比率 (Success rate)」、「服務滿意度 (Overall satisfaction)」、「計算任務使用成本比 (Overall cost)」、「徵詢非熟識服務次數比 (Rate of searching for unfamiliar service providers)」以及「誤用詐欺服務次數比 (Rate of misusing deception services)」。

在定義各個衡量標準之前，我們首先描述共通定義如下：

- 1) 計算任務成功完成：指得是一件計算任務取得所需要的相對應服務後返回原任務擁有者，且經確認沒有遭受服務詐欺。
- 2) 計算任務結束：指得是一件計算任務結束徵詢所需要的相對應服務過程後返回原任務擁有者，其結果是否已取得需要的服務，或是遭受詐欺等等並不在考慮之內。
- 3) 模擬時間 T ：我們令 T 指得是「自模擬之初的時間點 t_0 開始到現在時間點 t_1 間」的時間區間。
- 4) 滿意度最大值與使用成本最大值：我們對每一個任務擁有者所能給予信賴模型的「服務滿意度」以及「任務使用成本」設定了一個上限 (upper bound)，意即無論任何時間或任何計算任務，任務擁有者所能給予信賴模型的服務滿意度或是使用成本點數均不能超過此一上限。我們稱此上限為「最大滿意度點數」與「最大使用成本點數」。
- 5) 額外扣除點數：在我們的模擬之中，我們針對「服務滿意度」以及「任務使用成本」加入了額外扣除分數的要素：計算成功額外扣除 0 分、計算失

敗額外扣除 2 分以及計算結果為詐欺者額外扣除 5 分。這些額外扣除分數是由使用者根據經驗訂定，目前為了實驗模擬的目的而設定的適當值。任務擁有者會根據計算任務完成後的成果將額外扣除點數從「給予信賴模型的滿意度點數」扣除或者累加進「計算任務使用成本」之中。

- 6) 適當的服務：通常任務擁有者預期可以透過信賴模型快速地得到解決方案，無論此解決方案是不是所有方案裡面最優良的。所以我們定義所謂「適當的服務」指得是經由推薦制度所得到的優良或者是次優良的服務，且此服務必須符合計算任務所需。

我們利用從計算結果中所取得的資訊定義出五個衡量標準：

- 1) 計算任務成功比率：計算任務成功比率是測試案例主要的衡量指標之一。任務的委託與完成完全仰賴節點與節點之間信賴路徑的建立。信賴路徑愈快被建立起來，則任務擁有者將會愈快得到口碑優良的任務接收者所提供的服務；同樣地，服務品質優良的任務接收者也會因為信賴路徑愈快被建立，而愈快完成接踵而來的任務委託並累積口碑。如此一來，整個計算社會便會依照我們所預期的進入提升整體計算品質的良性循環。所以我們相信在相同的輸入及環境條件下，單位任務成功率越高，代表信賴路徑愈快被建立，使得任務得以越快被完成；任務越快被完成，代表著產能輸出就越好，信賴模型的效率也就越符合我們的預期。

我們定義 J_{total} 指得是「在模擬時間 T 中指派給動態模擬器的計算任務數目」， $J_{success}$ 指得是「在模擬時間 T 中成功完成計算的計算任務數目」，則我們可以得到「計算任務成功比率 (Success rate, SR)」為：

$$SR = J_{success} / J_{total} \quad \dots (6)$$

- 2) 服務滿意度：服務滿意度是用來平衡任務成功率的指標之一。我們將整個動態模擬器以及目前正在觀察的信賴模型視為一個「黑箱」(Black Box)，

因此除了高效率的產能輸出是我們關心的重點之外，我們亦關心任務在進入黑箱之後，是否如我們所期望地是因為信賴路徑的建立而被快速完成。我們將服務滿意度定義為任務擁有者對任務完成效率的滿意程度。在現實生活中，使用者對於服務品質的滿意程度往往取決於得到服務的速度、服務成果，最後是加上使用者個人的觀感給予綜合評價。

我們令「在模擬時間 T 中存在 m 個任務擁有者在其負責的計算任務結束時，所能給予信賴模型最大滿意度評價點數的加總」為 S_{max} 。假設模擬時間 T 中接受計算任務的 m 個任務擁有者所能給予信賴模型的最大滿意度評價點數分別為 $S_1, S_2, S_3, \dots, S_m$ ，則我們定義 S_{max} 如下：

$$S_{max} = \sum_{i=1 \text{ to } m} S_i$$

連續上述定義，令「此 m 個任務擁有者在其負責的計算任務結束時，給予信賴模型滿意度評價點數的加總」為 $S_{partial}$ 。假設 m 個任務擁有者給予信賴模型的滿意度評價點數分別為 $S_{p1}, S_{p2}, S_{p3}, \dots, S_{pm}$ ，其中， S_p 為 $S_{partial}$ 的簡寫。則我們定義 $S_{partial}$ 如下：

$$S_{partial} = \sum_{j=1 \text{ to } m} S_{pj}$$

則我們可以得到「計算服務滿意度 (Overall satisfaction, OS)」為：

$$OS = S_{partial} / S_{max} \quad \dots (7)$$

- 3) 計算任務使用成本比：計算任務使用的成本比率是一項從另一個角度來衡量信賴模型是否有效率的指標。我們將計算任務的成本簡化為信賴路徑的長度，也就是服務要求者在得到正確且優質的服務提供者服務之前的時間成

本。我們定義計算任務每經手一個節點，其計算成本便累增 1 點，當一個計算任務累積計算成本大於或等於計算成本最大值時，此計算任務無論處於何種狀態，均將被判定為失敗。整體計算任務使用成本愈低，代表信賴模型建立信賴路徑的方式愈有效率。

我們令「在模擬時間 T 中存在 m 個任務擁有者在其負責的計算任務結束時，所能負擔的最大計算任務成本點數的加總」為 C_{max} 。假設模擬時間 T 中接受計算任務的 m 個任務擁有者所能負擔的最大計算任務成本點數分別為 $C_1, C_2, C_3, \dots, C_m$ ，則我們定義 C_{max} 如下：

$$C_{max} = \sum_{i=1 \text{ to } m} C_i$$

連續上述定義，令「此 m 個任務擁有者在其負責的計算任務結束時，判定計算任務使用成本點數的加總」為 $C_{partial}$ 。假設 m 個任務擁有者判定計算任務使用成本點數分別為 $C_{p1}, C_{p2}, C_{p3}, \dots, C_{pm}$ ，其中， C_p 為 $C_{partial}$ 的簡寫。則我們定義 $C_{partial}$ 如下：

$$C_{partial} = \sum_{j=1 \text{ to } m} C_{pj}$$

則我們可以得到「計算任務使用成本比 (Overall cost, OC)」為：

$$OC = C_{partial} / C_{max} \quad \dots (8)$$

- 4) 徵詢非熟識服務次數比：除了有效地提升整體輸出產能外，我們亦關心信賴模型可能造成的「熱點瓶頸」問題，也就是服務瓶頸 (Hot-spots) [4]。隨著信賴模型推薦制度的建立，一段時間經過後，某個提供優質服務 α 的節點 A 會快速累積口碑，使得當節點 B 出現新服務需求 α 時，因為推薦制

度的關係，節點 B 將計算任務交付給節點 A 的機率便會大幅提升。倘若此時節點 A 已不堪負荷，如何縮短節點 B 重新尋找其他可提供相同服務的優質服務提供者的時間，便成為觀察信賴模型是否能有效率地重建信賴路徑的另一個重要課題。我們定義當節點 B 被迫重新尋找服務提供者時，倘若無法直接在自身或是朋友的使用經驗中取得另一個節點提供服務，則重新尋找服務支援的次數便累加 1 分。重新尋找服務支援累加的愈慢，代表從模擬開始之初一直到被迫重新尋找服務提供者的時間點 t 時間區間，信賴度的更新模式愈有效率，信賴路徑的重建亦更有效率，當然也就代表節點 B 重新尋找其他優質的服務提供者的時間也就會愈短，甚至可以直接從朋友中取得優良服務。

我們定義 U_{total} 指得是「在模擬時間 T 中指派給動態模擬器的計算任務數目」， U_{search} 指得是「在模擬時間 T 中計算任務因為無法從過去經驗得到適當服務，而必須徵詢非熟識服務的次數」，則我們可以得到「徵詢非熟識服務次數比」（Rate of searching for unfamiliar service providers, $SUSP$ ）為：

$$SUSP = U_{search} / U_{total} \quad \dots (9)$$

- 5) 誤用詐欺服務次數比：最後我們關心的便是伴隨信賴度更新而來的「排擠詐欺節點效應」。加速排擠詐欺節點，使詐欺節點因為受到小世界所有其他節點的公評，很快地無法接受到任務委託，對降低整體計算任務的失敗率以及避免計算任務成本浪費有很大的幫助。我們定義當計算任務回到任務擁有者 A 手上並判定受到節點 B 的服務詐欺時，誤用詐欺節點次數便累加 1 個誤用點數。

我們定義 M_{total} 指得是「在模擬時間 T 中指派給動態模擬器的計算任務數目」， M_{misuse} 指得是「在模擬時間 T 中計算任務經任務擁有者判定遭詐欺服務詐欺的次數」，則我們可以得到「誤用詐欺服務次數比」（Rate of misusing deception services, MDS ）為：

$$MDS = M_{misuse} / M_{total} \quad \dots (10)$$

- 6) 自我揭露程度：另一方面，我們希望可以用社會學的觀點來檢視模擬的結果，希望可以從模擬結果觀察得知，運用完整的社會學理論來實做的新信賴模型是否真的較傳統信賴模型優異。我們將社會學「關係發展理論」中，兩項用來衡量線上虛擬人際關係與現實生活人際關係差異性的指標性理論：「降低不確定性理論」以及「社會滲透理論」[30] 簡化並擷取兩者的主軸精神「自我揭露」應用到信賴模型上。所謂的「自我揭露 (Self-disclose)」就是進行互動的雙方會藉由增加溝通或揭露討論議題的寬度和深度，來達成對彼此更廣且深的理解。

關係發展理論認為，線上的虛擬人際關係之所以會比現實生活的人際關係來得發展快速、且更為複雜，最主要的因素便是虛擬的人際關係打破了時間與空間的限制，讓「虛擬的個體」不論是「自我重塑形象」（隱含本身原來的面貌）或是「推薦自我能力」（讓更多其他個體知曉進而進行下一步互動行為）都可以因為「口碑」（評價、推薦與信賴程度）的快速變化提昇在網路上的散佈程度。這正是我們期望看到的結果。

傳統的信賴模型僅僅以「朋友關係」做出發，這是一種較為單向的信賴度更新機制，事實上，我們可以說這樣的更新機制仍舊受限於「空間」因素。因為僅透過朋友關係來詢問服務，便是受到我們在第三章所提到的人際關係環境中的「第一區位」範圍的限制。相對地，認識其他優良的陌生人的機會便會大幅降低，當然就會拉長消弭熱點、尋訪到好友之外優質節點、或是利用整體小世界公評排除掉詐欺節點的時間。而本文所提出的新信賴模型引用社會學行動理論，讓節點之間有合作與競爭，於是好友名單便會常常接受「洗牌」，自然而然壓縮了消弭熱點與排擠詐欺節點的時間。我們將「自我揭露」程度簡化為「相互認識程度」，也就是將小世界中所有節點能保持的好友名單中的好友個數加總。在我們的預期之中，相互認識程度愈快被提高，信賴度的更新效率就會愈快被提升。

所謂的「好友」指得是節點因為計算任務的需要而透過推薦制度認識的節點。不論有沒有使用對方所提供的服務，或者是計算任務有沒有因此而完成，都算是「認識」了對方。在我們的預期之中並不希望小世界中的每一個節點記住所有的「曾經是好友，但是很久沒有再接觸過」的節點，或是「曾經使用過，但是很久沒有再去使用」的服務。主要的理由是如果每一個節點都可以擁有記住所有曾經交往的朋友或使用的服務，這不但不符合我們現實生活中的經驗（長時間沒有聯絡的朋友或服務，便會失去聯絡或遺忘），長時間下來使用信賴模型所建置的推薦機制便會失去意義。所以我們給予一個合理的「好友名單上限」（friend list upper bound），任何一個節點所能記憶住的好友數目必須小於或等於這個上限。

我們令「在模擬時間 T 中存在 m 個節點，將每一個節點的好友名單上限加總」為 F_{total} 。假設模擬時間 T 中一共存在 m 個節點，且 m 個節點所擁有的好友名單上限分別為 $F_1, F_2, F_3, \dots, F_m$ ，則我們定義 F_{total} 如下：

$$F_{total} = \sum_{i=1}^{m} F_i$$

連續上述定義，令「此 m 個節點已有的好友個數的加總」為 $F_{partial}$ 。假設 m 個節點已經認識的好友個數分別為 $F_{p1}, F_{p2}, F_{p3}, \dots, F_{pm}$ ，其中， F_p 為 $F_{partial}$ 的簡寫。則我們定義 $F_{partial}$ 如下：

$$F_{partial} = \sum_{j=1}^{m} F_{pj}$$

則我們可以得到「熟識率」（Acquaintance rate, AR ）為：

$$AR = F_{partial} / F_{total} \quad \dots (11)$$

下一節我們將呈現測試案例模擬的結果，並嘗試去提出解釋，藉以驗證本文所提出的新信賴模型。

6.4. 測試案例模擬結果

實際模擬的測試案例定義如下：

- 1) 整個測試案例包含了 20 個獨立節點，也就是我們令節點集合 Θ 為：

$$\Theta = \{ P_1, P_2, P_3, \dots, P_{20} \}$$

為了讓整個動態模擬環境更貼近真實分散式計算環境，我們讓每一個 Θ 中的節點都有「上線 (online)」、「離線 (offline)」、「當機 (crash)」、「重新啟動 (restart)」等四種狀態，且每一個節點發生任一個狀態的時間並非完全相同。

- 2) 整個測試案例中，一共會出現 18 種不同的服務，每個服務集合 \mathcal{E} 中的節點均會包含一個或一個以上的服務：

$$\mathcal{E} = \{ S_1, S_2, S_3, \dots, S_{18} \}$$

- 3) 我們設計了 100 件需求不同的觸發事件，包含不同的觸發時間，不同的服務需求，以及不同的承諾需求。每件定義於事件集合 ρ 中的觸發事件均為獨立互不相關的事件，且被指定的任務擁有者為隨機指定。唯一特別的是，我們在模擬的過程中，我們以觀察服務 1 為主要觀察標的，進行比較：

$$\rho = \{ E_1, E_2, E_3, \dots, E_{100} \}$$

除了這 100 件需求不同的觸發事件之外，我們還針對每一個參與模擬的節點設計了不同的四種狀態（上線、離線、當機以及重新啟動）時間。以我們的模擬過程為例，我們將擁有觀察標的服務 1 的節點概略分為數目相當的兩群，並且讓這兩群節點每隔 30 秒（即模擬數據中的時間點 4 及 8）便互換狀態。例如，在模擬時間經過 30 秒後，被分為 A 群中的節點狀態便會轉變為上線或重新啟動；而被分為 B 群的節點則會強制轉變為離線或是當

機。我們預期在稍後的實驗中可以觀察到不同的信賴模型對於不可預期的狀態變化有何應變能力。

4) 最後，我們定義每個節點僅可保留信賴度最高的 10 個好友：

$$\eta = \{ F_1, F_2, F_3, \dots, F_{10} \}$$

我們將前面所描述的「測試案例」在作業系統為 Windows XP 以及硬體配備為 Pentium® 4 CPU 2.8G Hz、1.00 GB Ram、硬碟大小為 80G 的電腦上執行模擬，每個不同的信賴模型均模擬十次，以取其平均值，並從模擬過程中截取 \log_2 (時間) 點上的數值做為記錄以及畫出曲線圖的依據。我們將每一個時間點的數據平均之後畫成曲線圖，定義曲線 *NT* 代表本論文所提出的新信賴模型，而 *DS* 代表 D-S evidence theory (D-S 證據推論模型)，最後我們用沒有使用信賴模型的機制 (No Trust，也就是僅依靠自身可提供的服務與朋友的服務來解決所擁有的任務。沒有任何推薦制度可用)，來做為評量的基礎。每一個公共生產力面向的評比描述如下：

1) 計算任務成功率：

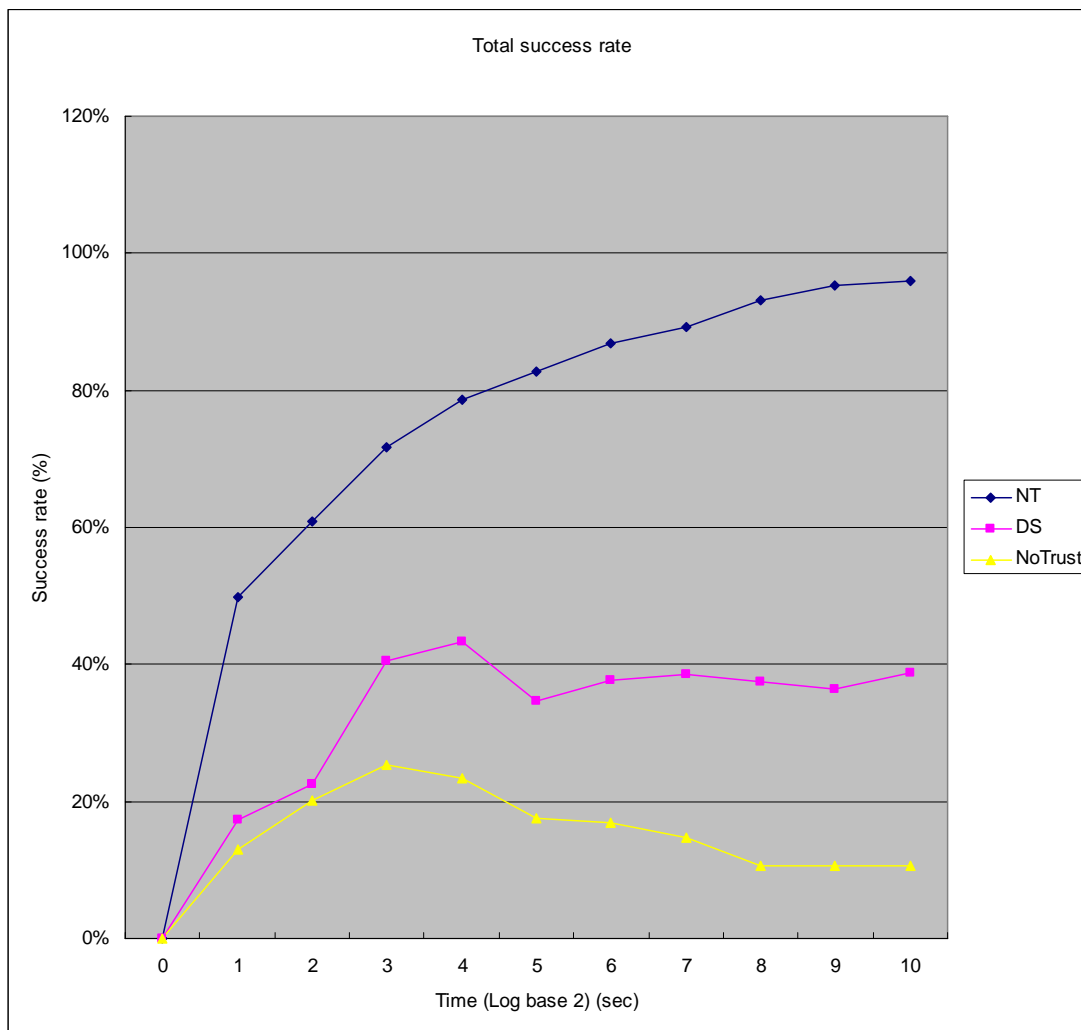


圖 6.6，測試案例模擬結果之計算任務成功率。

計算任務成功率顯示了計算任務透過信賴模型的推薦制度成功地得到適當服務的比率。我們可以從實驗數據中觀察得知，雖然傳統信賴模型因為擁有信賴推薦制度，使得計算任務成功比率一直維持在遠高於沒有使用信賴模型的數據之上，但是在擬真的動態分散式計算環境中，倘若遇到原先慣於使用的優質節點無預警的「離線」或「當機」，傳統信賴模型的反應速度卻與沒有使用信賴模型的數據一樣，因為無法快速地得知其他優良節點的存在，使得任務成功率嚴重下挫（時間點 4）。相反的，本文所提出的新信賴模型，因為有「合作」、「競爭」及「同業公會」機制的存在，使得信賴度更新率大幅地提升，強化了整體小世界面對無預警的離線或當機可以很快地找到另一個適用的服務。這也證明了新信賴模型的確在真實的分散式

計算環境中傳遞計算任務以及搜尋優質服務節點的優良能力。

2) 服務滿意率：

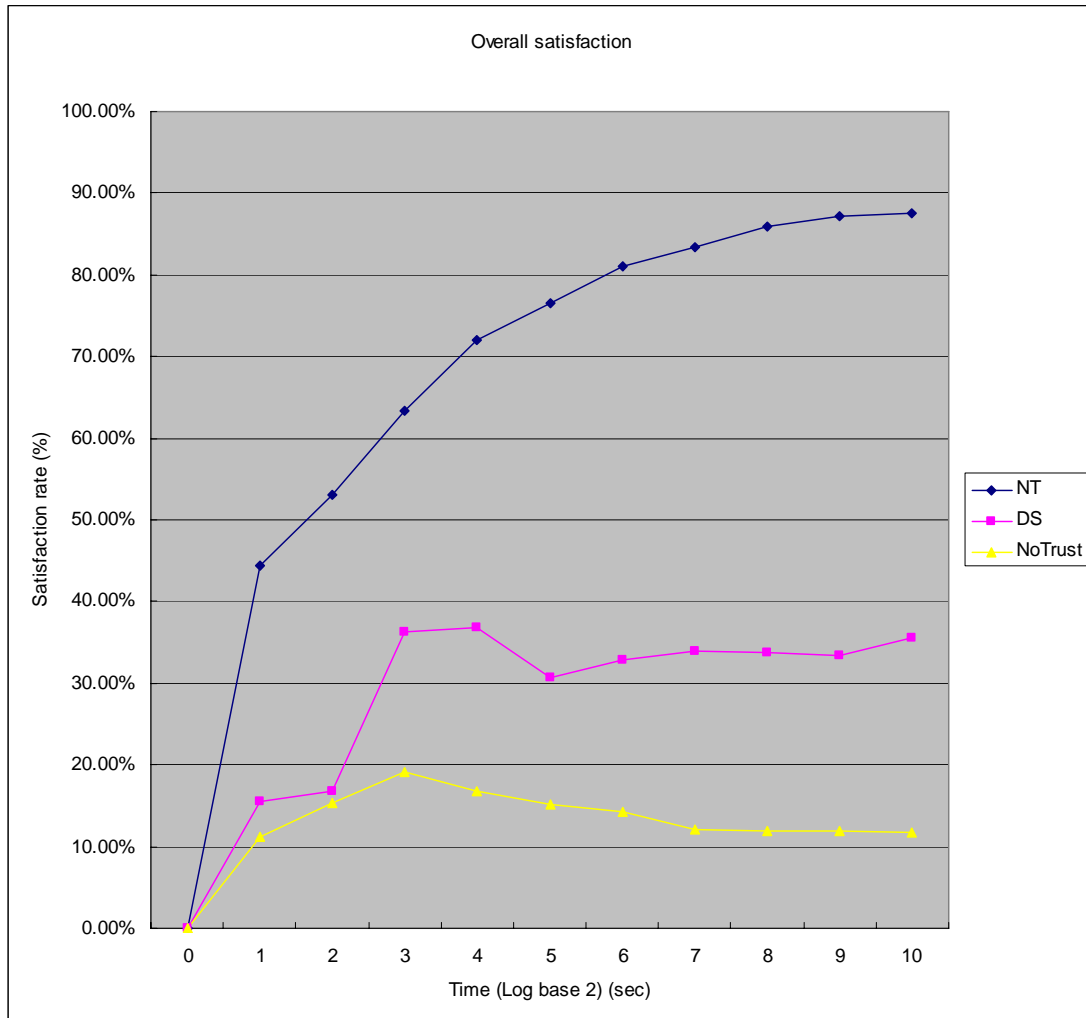


圖 6.7，測試案例模擬結果之服務滿意率。

服務滿意率顯示了擬真的分散式計算環境，對不同的信賴模型在推薦制度上的滿意程度。實驗數據顯示在時間點 4 的時候，傳統的信賴模型因為無法在無預警的節點離線後推薦另一個適當的服務途徑，使得整體小世界對其滿意度自時間點 4 之後便嚴重下挫，無法如本文所提出的新信賴模型一樣呈現緩慢的上升。回顧前一節我們對於服務滿意度的定義，我們認為一件計算任務的服務滿意程度跟被完成的速度與完成結果兩者的評分加總成正比，意即新

信賴模型在計算任務取得優質服務的「正確」以及「效率」兩個面向上可以有較為優異的表現，也證明了本文一開始對新信賴模型所定義的人際關係環境與社會行動理論，的確可以很快地提升信賴模型的穩定性。

3) 計算任務使用成本比率：

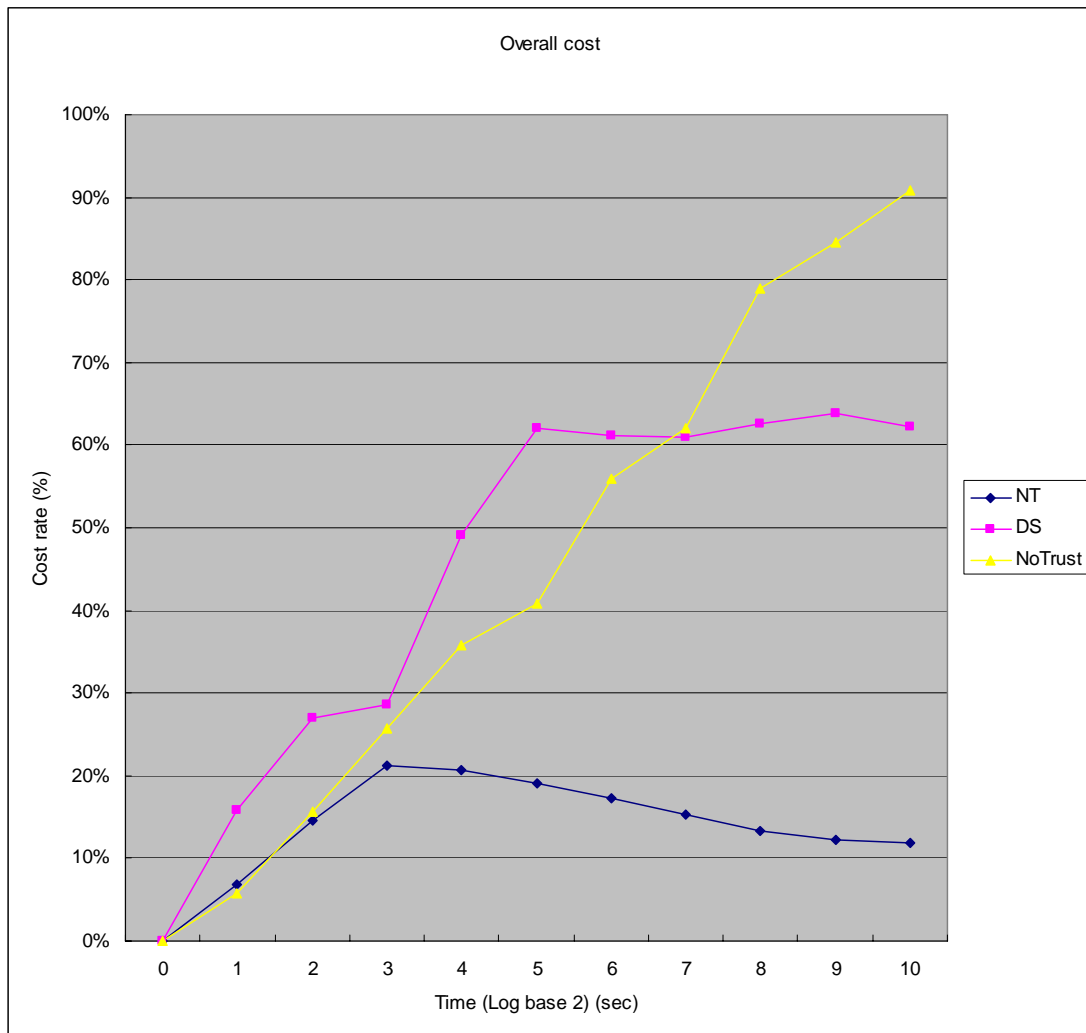


圖 6.8，測試案例模擬結果之計算任務使用成本率。

計算任務使用成本顯示了計算任務自指派之初到取得服務之前所累加的使用成本。沒有使用信賴模型的數據一直居高不下，且不斷向上攀升；而傳統的信賴模型則是在面臨無預警的節點離線時呈現巨幅的成本增加，這意味著

D-S 證據推論模型在信賴路徑的建立上效能上仍有改進的空間；而本文所提出的新信賴模型恰巧利用了同業公會制度提升了這一方面的信賴度更新速度，使曲線因為小世界中節點間來往的密集程度的提升逐漸地遞減並趨於一個平穩的狀態。

4) 徵詢非熟識服務次數比：

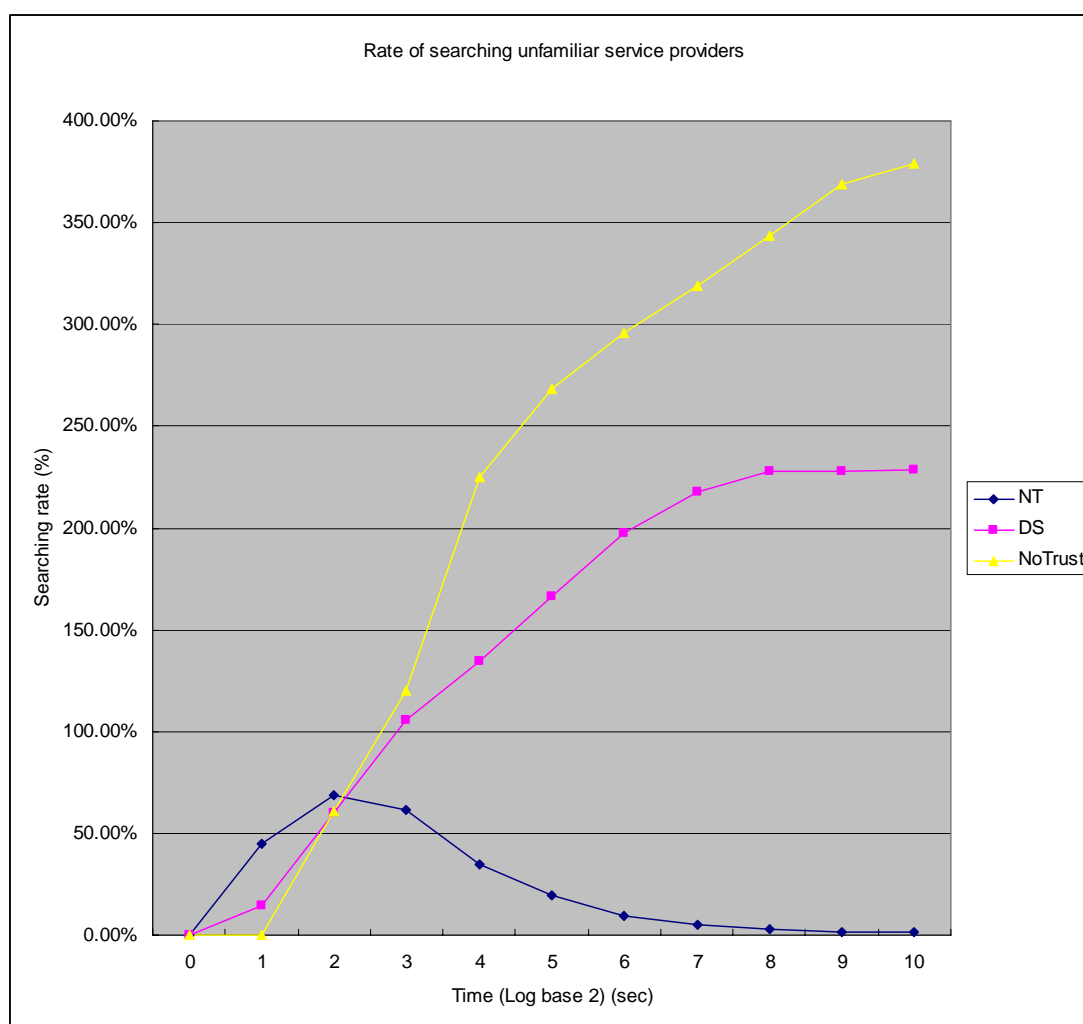


圖 6.9，測試案例模擬結果之徵詢非熟識服務次數比。

徵詢非熟識服務次數比率顯示了計算任務在被迫重新尋找並取得一個優質服務提供者的服務之前所必須尋問的次數。新信賴模型利用社會學行動理論的合作與競爭模式，建立起同業公會的人際關係環境，增加了「結識好友」的

速度（也就是我們所引用的社會學關係發展理論裡的「降低不確定性理論」與「社會滲透理論」），使得從模擬之初新信賴模型所呈現出來的曲線便很快地高於傳統信賴模型的曲線，並不斷地上升；之後因為信賴度更新效率的優異表現，讓曲線趨於收斂，表示需要再次徵詢非熟識服務的現象大幅下降。這也顯示了因為我們引用了完整的社會學理論，使得結識好友的速度提昇非常多，以至於尋找其他非熟識的節點提供服務的現象提早出現。

5) 誤用詐欺服務次數：

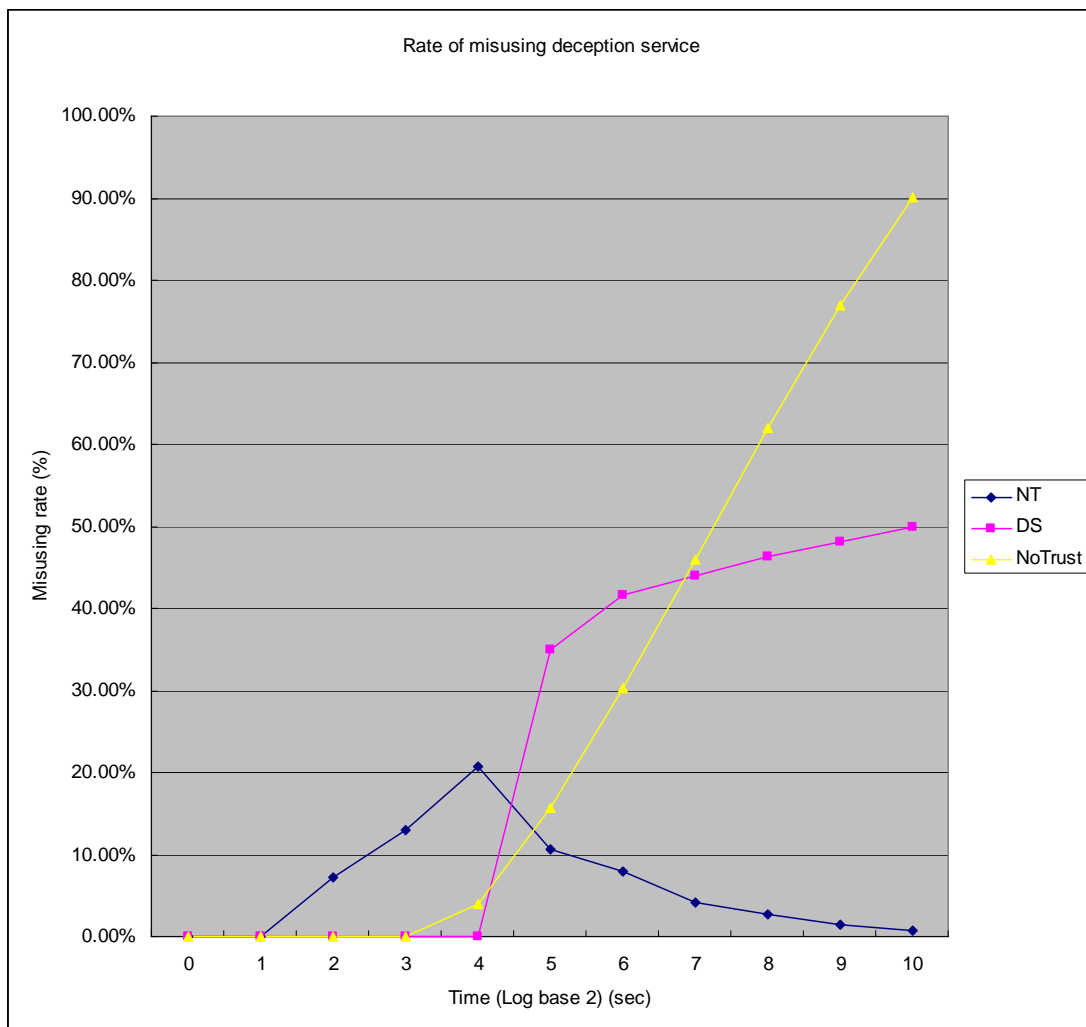


圖 6.10，測試案例模擬結果之誤用詐欺服務次數。

我們透過誤用詐欺服務次數曲線圖，驗證本文所提出的新信賴模型在排擠詐

欺節點效率上的表現。如圖 6.10 所示，*NT* 誤用詐欺服務曲線所顯示的誤用次數，從時間點 4 之後慢慢地趨於緩和，並且呈現收斂的趨勢；而 *DS* 曲線仍始終不斷累增誤用次數，最後仍無法有效地降低誤用詐欺節點的機率。這顯示本論文所提出的新信賴模型確實有能力在短時間之內將提供詐欺服務的節點逐漸排除在外，並透過信賴度快速地更新，提升了整體小世界在分散式計算環境中的計算品質。

6) 自我揭露程度高低：

最後我們透過社會學的角度來檢視本文所提新信賴模型是否因為引用了完整的社會學以及人際環境關係的定義，而在信賴度的更新效能上有所提升。根據社會學關係發展理論的定義，人際關係的發展過程速度的快慢，取決於個體自我揭露程度的高低。自我揭露程度愈高，整體小世界的「降低彼此的不信任感」（降低不確定性理論）與「將彼此的關係往更親密的方向提升」（社會滲透理論）兩個指標性程度便會愈高，自然整體小世界的計算品質與產能便會愈好。我們定義將兩個信賴模型中每一個單一節點的好友個數加總並且取百分比為「熟識率（Acquaintance rate）」。在我們模擬的測試案例中，我們配合前面所分析的「計算任務成功率」的曲線圖我們把分析結果繪成如下圖 6.11：

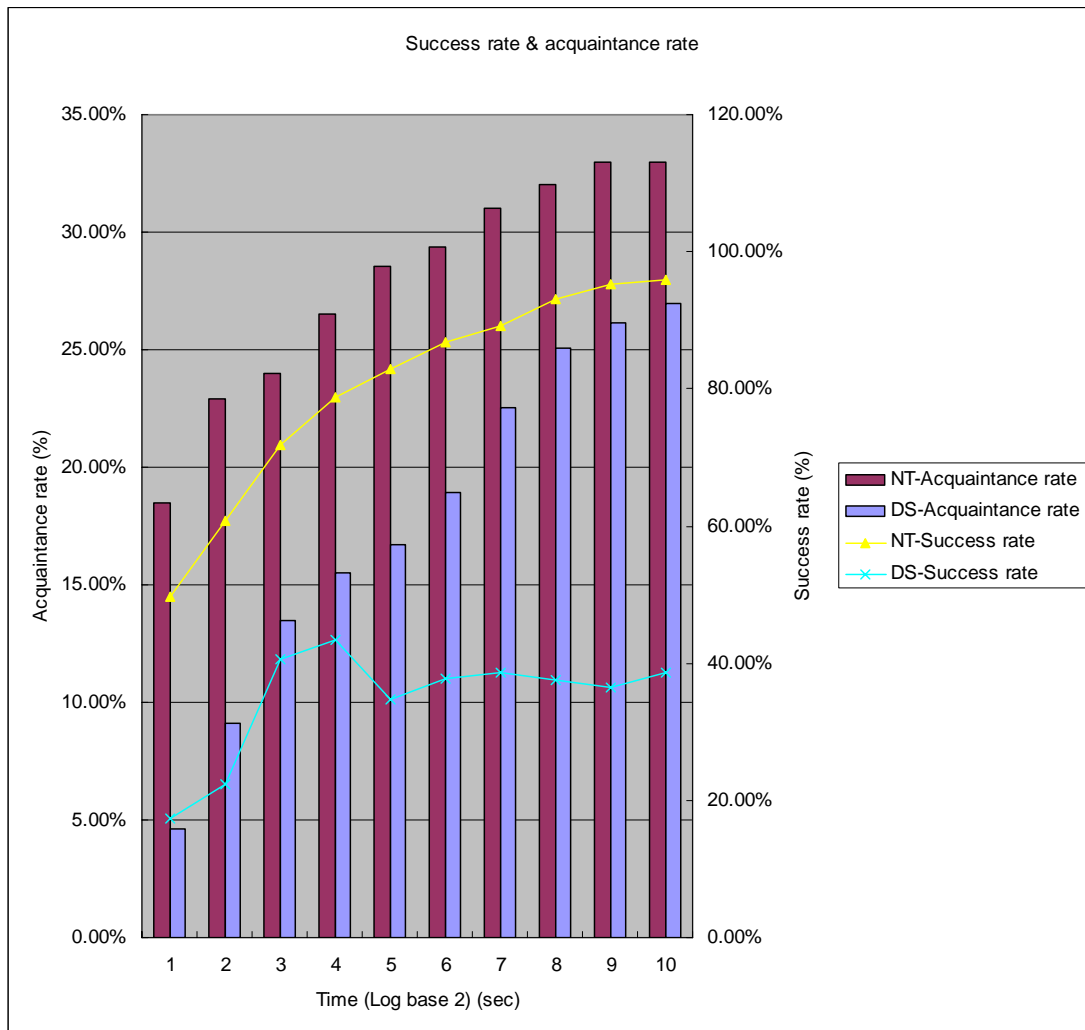


圖 6.11，測試案例模擬結果之自我揭露程度。

我們從圖 6.11 可以看出，新信賴模型因為採用較為完整的社會學理論，所以在自我揭露程度（也就是 Acquaintance rate）的成長上明顯地較 D-S Evidence Theory 模型來得快。自我揭露度提升得快，相對的降低不確定性以及社會滲透兩方面自然比 D-S Evidence Theory 來得好，信賴值更新的效率也就好很多。這與我們之前的預期可說是十分的吻合，也證明了本文所提出的新信賴模型確實因為引用社會學行動理論，讓整體分散式計算環境的品質與效率提升許多。

綜合上述五個公共生產力面向的評比，本文所描述的新信賴模型，基於社會

學的行動理論與人際關係環境理論所建立起的新信賴關係，確實提昇了整體小世界的公共生產力並對節點的詐欺行為出現有效的排擠效應，讓小世界中的每一個節點都能趨良逐劣。我們認為，新信賴模型中仍有許多值得改進的地方，倘若未來的研究之中可以加以改良，甚至是更加貼近實際的人際環境關係，對信賴機制在分散式運算環境上的應用將會提供更大的助益。

