

設計 NS2 分析模擬之整合模擬環境

Design a Integrated Simulation Environment for NS-2 simulation Analysis

李龍盛
國立嘉義大學資訊工程學系
助理教授
e-mail :
sheng@mail.ncyu.edu.tw

許惠君
國立嘉義大學資訊工程學系
大學部學生
e-mail :
s0942777@mail.ncyu.edu.tw

周千楓
國立嘉義大學資訊工程學系
大學部學生
e-mail :
s0942778@mail.ncyu.edu.tw

摘要

對一網路環境做效能分析時，最簡易的方法即是使用網路模擬工具。基於開放原始碼及免費軟體，NS-2 網路模擬器常是網路實驗上的主要選擇。但 NS-2 並非商業軟體，並無一套完善操作介面讓使用者可以輕易上手，在進行模擬實驗時需花費大量的時間和勞力去一步一步設定及建置模擬環境。NS-2 實驗程序主要分成三大部分：建置環境而模擬、分析實驗數據及繪出實驗分析圖表。在此文章中，我們提出一個整合模擬環境進而達到整合實驗三大步驟之所有需要一步一步操作的複雜步驟，並且在此文章中證實此整合模擬環境之效能比原本的方法更為佳。

關鍵詞：NS-2、效能、整合

Abstract

To analyze the network performance, the easiest way is to use a simulation tool to simulate the network. NS-2 is the most popular tool for the network simulations. The procedure of the simulation analysis is not simple. We can partition the procedure into three parts: the simulation runs the numerical analysis, and the analysis graph drawing. We propose an integrated simulation environment to integrate the three parts of the simulation analysis procedure. We also show that the integrated environment outperforms the traditional development environment.

Keyword : NS-2、Performance、Integrate

1. 前言

現今在一產品問世之前，或是架構實際網路之事前準備工作，需先經由網路模擬器 (Network Simulator) 來模擬操作整個環境，可預先驗證演算法效能及缺失等。因網路模擬器能在架設實際環境之前先模擬該網路環境，藉此來查看實際上可能會發生的問題而加以改善。所以使用模擬器的效率好與否，也是選擇上的限制之一，而目前最廣為使用的網路模擬器為 NS-2 [1]。

NS-2 (Network Simulator, version 2) 是一套物件導向的網路模擬器，它能模擬真實網路系統的架構與特性，C++ 與 OTcl 為其開發語言。而 NS-2 能模擬有線網路或無線網路等等相關的實驗環境，額外能配合其他工具匯出圖表，也可使整個網路環境動向視覺化。但 NS-2 並不是一個容易入門的工具，使用過程之複雜，所以本文將提出一套作法來簡化使用過程。在此將簡述動機、原來的方法、改善後的方法、此文所使用的做法、實驗結果、比較及結論。

1.1. 動機

NS-2 的整個模擬實驗過程從開始到匯出數據、繪出圖表需要相當繁複的過程，往往前置作業就需花費相當多精力及耗費相當多時間，然而一次只能執行模擬一次，但實驗數據的需要量卻是龐大的。使用模擬工具即是為了能先了解該網路環境，如此不便，會降低使用模擬工具的其效益。使用方法上之不

便，再加上 NS-2 非容易入門之工具，使用上就更加困難了。而本篇文章，即是針對這些問題做改善，額外比較、分析改善的時間等問題。本文範例主要是著重在 Ad Hoc network 部份，使得使用這網路模擬工具來做網路實驗時，能更加方便、更有效益。有效率地使用此模擬器，對真實網路環境也是一大美處。

1.2. NS-2

NS-2 是一套物件導向的網路模擬器，由 UC Berkeley 所開發完成，C++與 OTcl 為其開發語言。而使用者使用的方式採用 TCL(Tool Command Language) 為其腳本語言(script language)，撰寫所需模擬的網路環境和參數設定等。在網路架構上它可模擬 router、link、end point 等，在網路情況中可以模擬封包延遲、封包丟棄、封包等待於佇列等 [2]。

NS-2 之下除了自己擁有的許多工具之外，如無線網路工具：cbrgen 等，可再配合 linux 下之繪圖工具、動畫工具等使功能更完善[4]。

1.3. NS-2 實驗步驟簡介

NS-2 之一般實驗方法為先對所要模擬的網路環境撰寫一個 TCL 語法的有線或無線網路主程式；使用 NS-2 模擬之後，模擬中的整個過程之紀錄將會全部紀錄在一個檔案裡，稱之為 trace file(tr 檔)。而使用者即可利用此 trace file 來計算出使用者所想要的實驗結果，如傳送封包之 throughput 或封包遺失率等，可利用 Linux 底下之 awk 工具來計算出這些用來評估效能的結果。而觀察實驗結果最直接的方式即是利用圖表的方式，則可使用 Linux 底下之另一工具 gnuplot 來匯出實驗圖表，這樣即可大致完成一個有線網路實驗[5]。

而無線網路實驗的部份，尚須考慮節點會移動的情況及封包傳送的情形。考慮一個適當的無線網路環境，節點會隨機移動，封包的傳送也應是隨機，則 NS-2 底下提供兩個工具 cbrgen 及 setdest 即可針對此部份做模擬，完成無線模擬。

1.4. NS-2 模擬器與相關工具

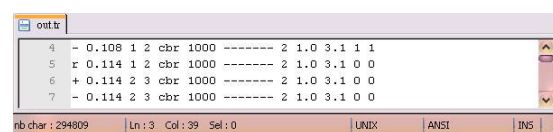
此節介紹 NS-2 的工具，NS-2 無線網路工具之 cbrgen，它的功能為產生無線網路中的

CBR 或 TCP 隨機資料流，其用法為輸入(1)輸入 TCP 或 CBR 資料流(2)指定節點數(3)亂數種子(4)最大連線數(5)CBR 資料流中每秒送出幾個封包。另一無線網路工具 setdest，其功能為隨機產生無線網路模擬所需節點移動場景，可在指定的矩形範圍內產生指定節點數，節點會朝隨機指定的節點前進移動，並決定停留時間。其用法為輸入(1)節點數(2)停留時間(3)節點隨機移動最大值(4)時間(5)移動範圍之 X 軸(6)移動範圍之 Y 軸 [3]。

而可配合 NS-2 模擬實驗的 Linux 底下之工具，首先介紹處理 NS-2 模擬記錄檔的 awk 工具，它是一種使用直譯器的語言，善於處理記錄檔或欄位型態資料的工具。我們使用它來處理模擬後產生的 traffic trace file 的資料，做數值分析之後成為我們需要的效益評估參數。而觀測實驗結果最簡潔的方法即是利用圖表呈現，使用 awk 工具分析出來的數據，可利用繪圖工具 gnuplot 來產生實驗圖表。Gnuplot，一個命令導向的交談式繪圖程式(command-driven interactive function plotting program)，需逐步設定、輸入來修改繪圖環境，其輸入格式為座標檔。額外可搭配 NS-2 之工具 nam，可視覺化整個網路的模擬過程，對初學者甚有幫助，但對使用 NS-2 來做網路模擬實驗的人，是可選擇性使用的工具。

除了這些重要工具之外，最重要的莫過於模擬實驗後產生的 traffic trace file，它記錄了整個模擬實驗的每個動向、步驟，使用者可藉此來評估實驗結果，如表(1)[6]。

表(1)中第一欄 event 指的是在第二欄 time 的時間下發生的事件，若是“r”即是收到封包，“s”表送出封包，“+”為加入等待的 queue，“-”移出 queue。第三欄為此事件的動作者之 source node，第四欄為被傳送的對象—destination node，傳送封包的類型為第五 (packet type)、六(flag size)、七(flag id)、十一欄(packet id)，八、九欄為傳送兩邊之 port，第十欄則為此事件的 id，如圖(1)。



Event	Time	Source Node	Destination Node	Packet Type	Flag Size	Flag ID	Packet ID	Port 1	Port 2	Event ID
4	- 0.108	1	2	cbr	1000					----- 2 1.0 3.1 1 1
5	r 0.114	1	2	cbr	1000					----- 2 1.0 3.1 0 0
6	+ 0.114	2	3	cbr	1000					----- 2 1.0 3.1 0 0
7	- 0.114	2	3	cbr	1000					----- 2 1.0 3.1 0 0

圖(1)：一列代表一事件，如送出封包、封包加入或移出 queue、接收封包等。

1	2	3	4	5	6	7	8	9	10	11
event	time	S. node	D. node	P. type	F.size	F.id	S. port	D. port	Seq. N	Pkt.id

表(1)：trace file 記錄格式

2. 原 NS-2 進行 Ad-hoc 實驗之方法

2.1. Ad-hoc Network

無線隨意網路(Wireless Ad Network)中節點不須經由無線基地台，而是以對等的方式對網路進行存取。其路由協定分成兩大類：
 (1)Proactive routing protocol—每個無線節點固定一段時間就會發出路由資訊，並且藉由從其他節點接收到的路由資訊不斷更新自己的路由資料表，而達到隨時有一份最完整的路由表，使封包傳送時即知道目的路徑。
 (2)Reactive routing protocol—無線節點只有在需要傳送封包卻找不到路徑時才開始運作，即來源節點會呼叫路徑發現程序(route-discovery process)。因此此協定不需要保存完整路由表，所以需要的頻寬較小，但無法即時找到目的路徑 [7]。

2.2. 評估效益參數

效益評估參數為針對模擬後產生的實驗過程紀錄進行分析、計算，經由一系列的運算成為我們評估實驗結果的重要依據。底下為我們實驗中所需的評估效益參數：

(1)封包送達比例，公式為：

$$\left(\frac{\text{received Packet 總數}}{\text{send Packet 總數}} \right) \times 100$$

從 trace file 中計算每一列的第一欄 event，r 即為接收封包的事件、s 為送出封包的事件，各別計算 s 及 r 事件的總和。計算接收封包總數除以送出封包總數乘上 100 即為封包送達比例。

(2) 封包平均 end-to-end delay 時間，包含所有可能延遲時間的總和，如發現路徑的緩衝時間，mac 層的重傳時間，傳遞時間等。公式

為：

$$\left(\frac{\text{end to end delay time 總和}}{\text{received Packet 總數}} \right)$$

(3) 第一個封包收到的時間，用來評估路由表收斂的時間。

2.3. 一般 Ad-hoc 實驗方法

操作 NS-2 網路模擬實驗時，需要一連串的步骤，原本的方法從建構一個網路環境到產生實驗數據，再經過複雜的運算繪出圖表，需要相當繁複的過程，在此介紹一般無線網路的實驗方法。首先，執行 cbrgen 這個工具來產生 TCP 或 CBR 隨機資料流檔案，執行時需給 cbrgen 所需要輸入且使用者所指定的參數來產生檔案。再執行 setdest 工具，同樣在執行時給予參數而產生隨機網路之節點隨機移動場景。並將上述的資料流檔案及節點移動檔案匯入所寫好的 tcl 語法的無線網路主程式中，以便執行模擬。使用 NS-2 執行模擬之後，將會產生實驗總紀錄之 traffic trace file 及可供觀察模擬動向的 nam 檔。再寫一個 awk 程式，從 traffic trace file 中挑選各種計算效益評估參數時會使用到的紀錄來計算。使用 awk 工具執行分析過後，輸出資料，再利用 gnuplot 繪圖工具將 awk 輸出的檔案匯出為圖檔，以此實驗結果來做整個網路環境之影響等來作評估。而想要產生一個好的圖檔，卻要輸入許多 command 才能達成。以下為操作交談式繪圖程式 gnuplot 之相關步驟：

```

Terminal type set to 'unknown'
gnuplot> set title "output"
gnuplot> set xlabel "times"
gnuplot> set ylabel "delay time"
gnuplot> unset key
gnuplot> plot "output" with linespoints_

```

圖(2)：設定圖檔 X 座標 label 及 Y 座標 label 過程

```

Send bugs, suggestions and mods to <info-gnuplot-beta@dartmouth.edu>

Terminal type set to 'unknown'
gnuplot> set terminal gif
Terminal type set to 'gif'
Options are 'small size 640,480'
gnuplot> set output "output.gif"
gnuplot> replot

```

圖(3)：存為 gif 檔

```

~/exp-1
gnuplot> set xlabel "times"
gnuplot> set ylabel "packet delivery fraction"
gnuplot> set terminal gif
Terminal type set to 'gif'
Options are 'small size 640,480'
gnuplot> set output "output.gif"
gnuplot> plot "out.dat" using 1:2:3 with errorbars 1, "out.dat" with linespoint
s 1

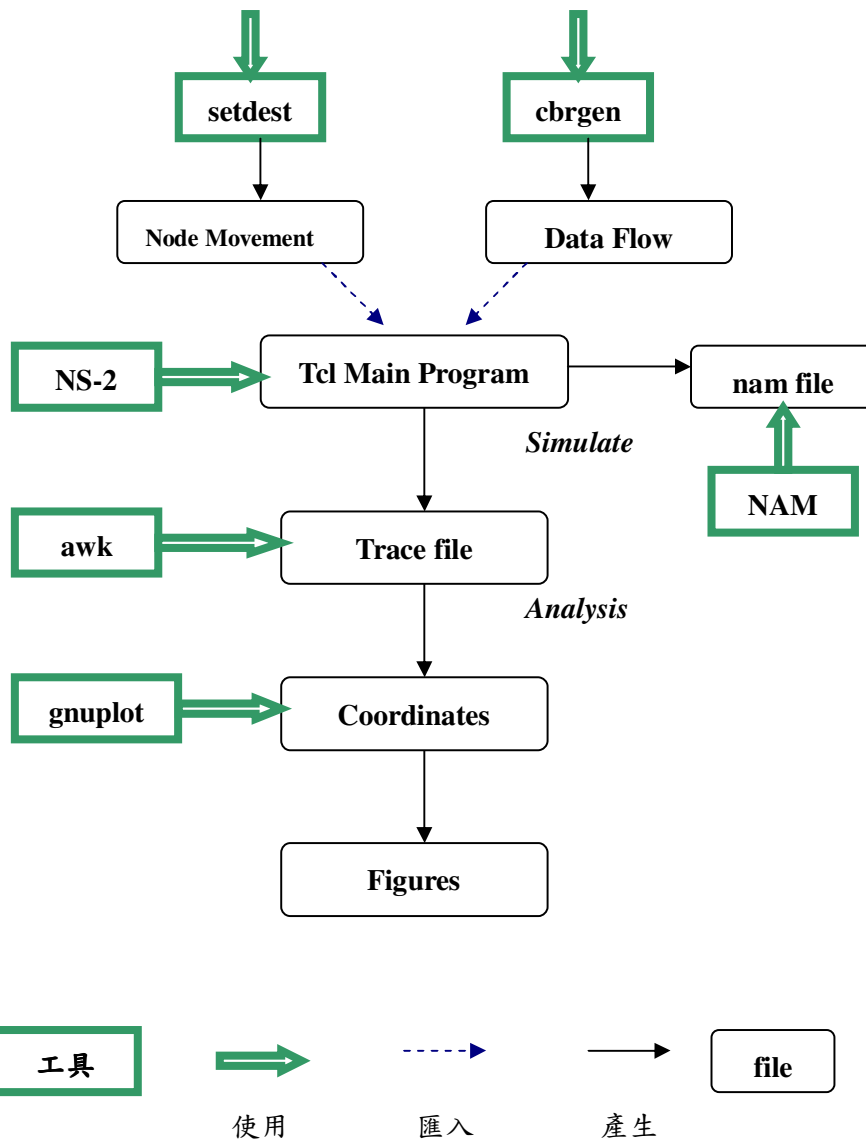
```

圖(4)：計算完的評估參數需要繪成信賴區間 (CI)圖檔

製作一個圖檔需要輸入上列等許多指令，非常繁複，如此，執行一次模擬所需要花的時間實在是太大了，但對於一個需對網路環境

評估的網路模擬實驗，模擬結果數據所需要的實驗次數動輒數百次、數千次，而整個實驗過程又如此複雜且 NS-2 本身不夠容易上手，降低了模擬工具的效益。

底下為一般方法之流程圖，粗方框代表 NS-2 工具或其他工具，粗箭頭為對某個檔案使用了那種工具，虛線箭頭為 A 檔案匯入到 B 檔案，細直箭頭為產生檔案，細方框為檔案。一開始分別使用 setdest 與 cbrgen 工具，產生 node movement 檔案及 data flow 檔案。將這兩個被產生的檔案匯入到以 TCL 語法寫成的 NS-2 的無線網路主程式，再使用 NS-2 執行模擬，將產生 trace file 及 nam file。再使用 awk 工具及寫好的 awk 程式對 trace file 做數值分析成評估效益參數，最後再使用 gnuplot 繪圖工具繪出實驗圖表。



圖(5)：原方法流程圖

3. 改進實驗方法

3.1. 進行改進的方式

本文採用撰寫一 shell script 程式來包覆所程式和步驟，只需執行此程式，執行後將出現一使用者介面，調好參數即能等待最後結果。

我們先在 tcl 無線網路主程式裡，使它能自動執行 cbrgen 及 setdest 這兩個工具。並且把 shell script UI 輸入的參數，將它傳送到 cbrgen 及 setdest 工具裡，使執行主程式時即能一同執行 cbrgen 及 setdest。

將各種計算評估效能參數的 awk 程式及繪圖的 gnuplot command 都分別寫好成各個檔案。再寫一個 perl 語言的 shell script 程式來包覆所有的程式並且我們提供一個使用者介面來作為使用者與工具之間的橋樑，則只需執行這個 shell script 程式，就能執行完所有的步驟，實驗結果即產生。

執行 shell script 時先產生 UI 使用者介面，直接在此選擇各種使用者所需要的參數，並且可以選擇延續上次實驗的數據繼續實驗，或是全新的實驗。shell script 程式就會開始呼叫及執行各個工具，並把上述提到的各種工具所需要的參數傳送過去，且根據使用者所選擇的次數來執行。最後會產生實驗結果、數據及圖表，且使用者可根據 trace file 或計算出來的數值來檢查實驗過程或做修改依據，使用者也可以根據此來 debug，另外可以執行 nam 程式來觀看整個模擬實驗其節點傳輸及移動場景的動畫。

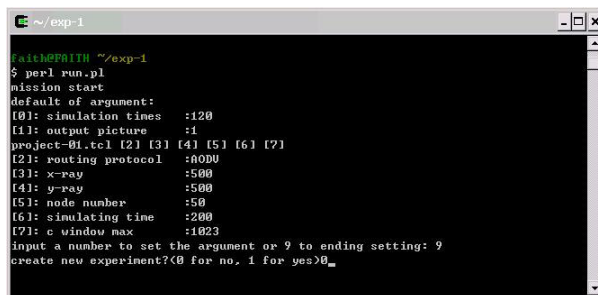
除此之外，我們尚改善 NS-2 的小工具，以達到實驗正確性增高。像是 cbrgen 工具，在隨機無線網路中，資料流應該也是隨機的。而在 cbrgen 工具中亂數種子卻不能選擇隨機的種子(0)，需要進去 cbrgen 裡將判斷式改成 seed < 0 才能選擇此種子。

使用者也可仿本文所使用的方法，增減自己所需要的功能以可適用其他網路實驗，而有些參數不在本文所使用範圍內的話，只

需到主程式修改此參數即可。

3.2. 改進後的 Ad-hoc 實驗之方法

改善之後，採用執行一 shell script，執行後會出現使用者介面，選擇自己所需要的參數之後，開始模擬，結果將自動匯出。使用步驟為先執行採用 perl 語言寫的 shell script，此時程式會給一個需輸入參數的使用者介面，選擇自己所需要的參數，如：次數、protocol、拓樸大小、節點數、評估效益圖、模擬時間長度、c window max，一開始這些參數已經有設定預設值，有要變動的部份，再選擇那個參數調整成使用者想選定的大小。選好參數之後，開始執行模擬，等全部的實驗跑完，最後 traffic trace file、nam file、figures 會全部產生。另外，traffic trace file 可供 debug 時使用。

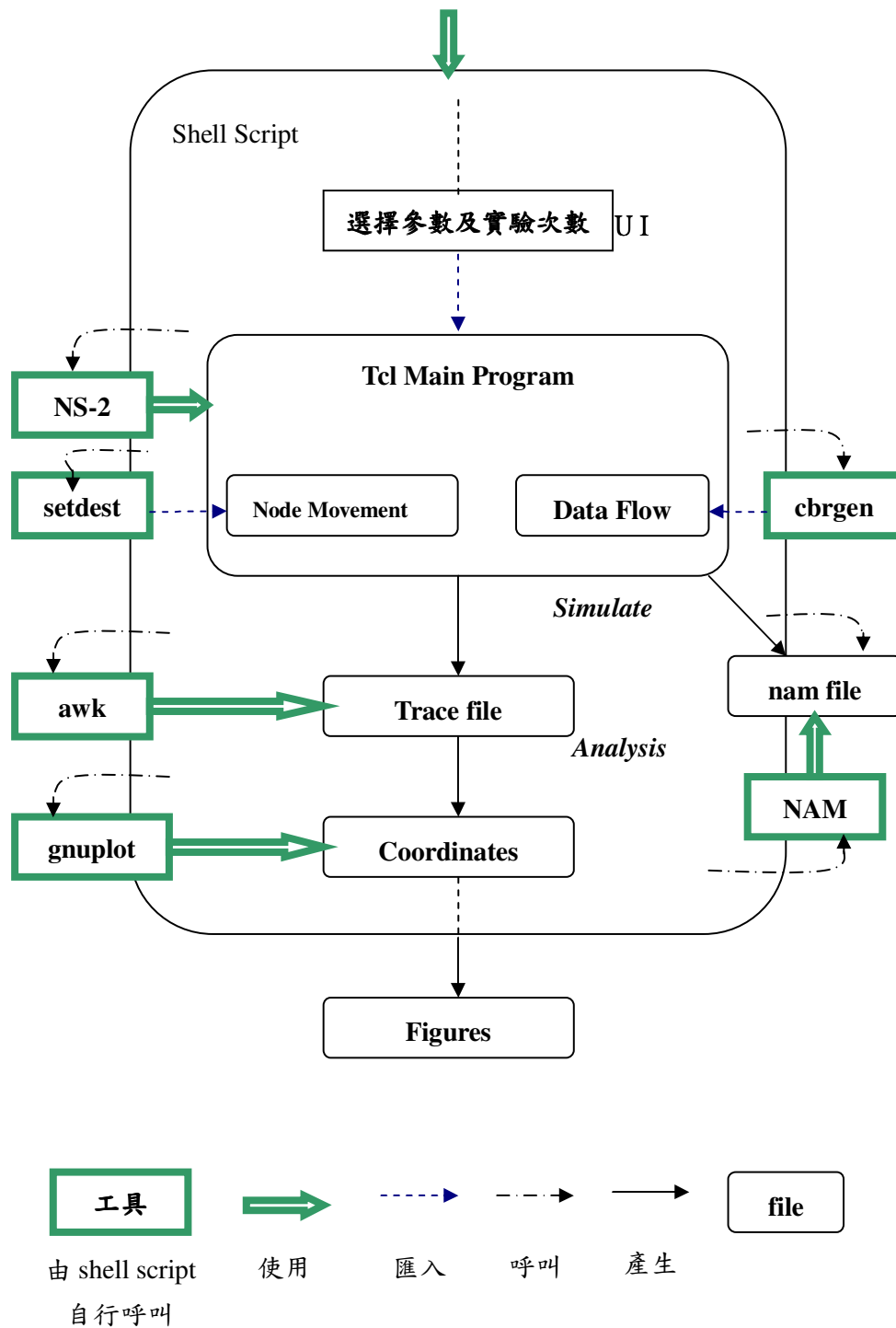


```
~/exp-1
$ perl run.pl
mission start
default of argument:
[0]: simulation times :120
[1]: output picture :1
project-01.tcl [2] [3] [4] [5] [6] [7]
[2]: routing protocol :0000
[3]: x-ray :500
[4]: y-ray :500
[5]: node number :50
[6]: simulating time :200
[7]: c window max :1023
input a number to set the argument or 9 to ending setting: 9
create new experiment?(0 for no, 1 for yes)0_
```

圖(6)：執行 shell script 後的 User Interface

藉由輸入代號去改變預設值成使用者所需要的參數大小，最後選 9 結束設定，並且選擇新的實驗或延續上次的實驗數據。(0)模擬次數、(1)圖、(2)protocol、(3)拓樸 X 之大小、(4)拓樸 Y 之大小、(5)節點總數、(6)模擬網路時間、(7)c window max。

底下為改善後的方法流程圖，與一般方法流程圖類似，差別在於粗方框為工具，但是為 shell script 所呼叫，非使用者自己執行。使用者只在一開始執行 shell script 程式，接下來產生使用者介面，選定好參數之後，shell script 程式會在各個適當時間呼叫各個所需執行的程式，直到模擬結束，直接產生實驗圖表。



圖(7)：新方法流程圖

4. 實驗與分析

4.1. 模擬環境

本模擬實驗是使用安裝在 Linux 的 NS-2.32 進行 Ad-hoc 無線網路的模擬實驗，實驗所使用的路由協議為 AODV。實驗中的無線網路節點數有 50 個，移動空間為 500m*500m，節點移動最大速度為 10m/sec，

節點在到達目的地前不會有停留動作。傳送的封包是 CBR 封包，封包大小為 512 bytes，每秒傳送 10 個封包。

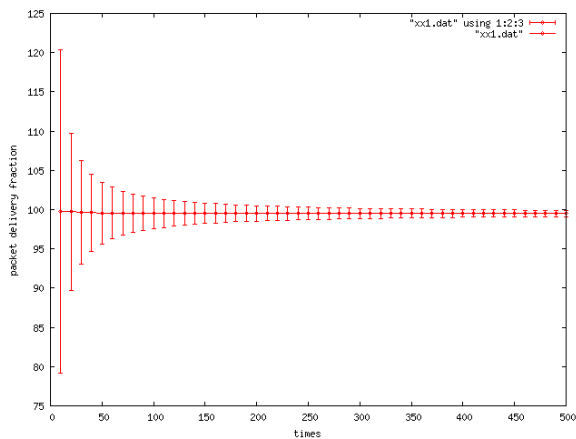
第一組的實驗是以每次 200 秒進行 500 次的模擬後，每 10 次計算一次統計資料的平均值。第二組實驗進行一次 200 秒的模擬後，每 10 秒計算一次統計資料的平均值。

4.2. 模擬結果與分析

在此藉由比較(1)多次實驗中的各個評估效益信賴區間圖及(2)單次實驗中各個時間點之評估效益參數圖表來分析、比較此文章所提出之整合模擬環境的實驗結果。

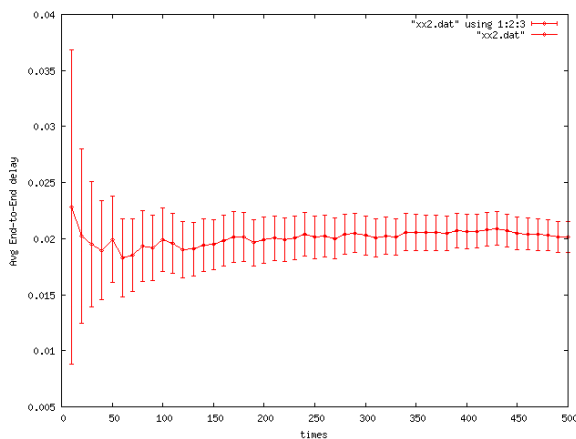
4.2.1. 針對大量實驗次數統計

此小節實驗為藉由大量實驗次數來分析新方法之實驗結果，來分析實驗之可靠度，包含封包送達比例、平均點對點延遲及平均第一個封包送達時間。



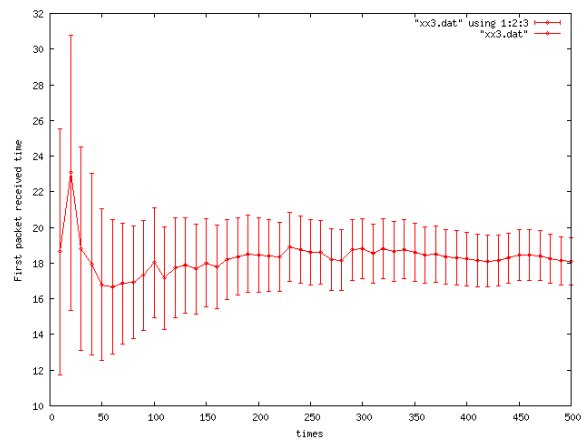
圖(8)：封包送達比例 CI 圖

藉由此圖可觀察出，實驗次數愈多，數據愈趨於穩定，是一個十分穩定的圖。



圖(9)：平均點對點的延遲時間 CI 圖

此圖在實驗次數較少時，波動幅度很大，但在實驗次數增多後，也漸漸趨向於穩定的狀態。



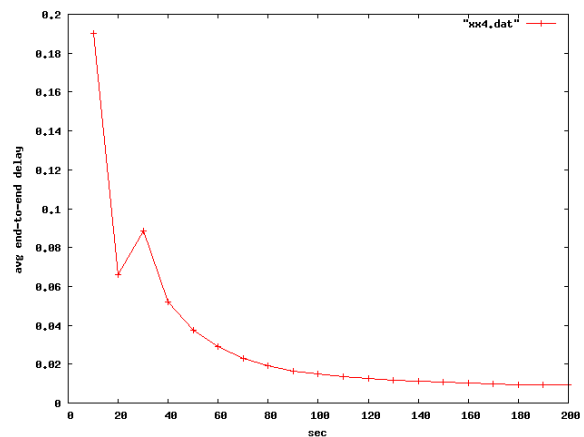
圖(10)：平均第一個封包送達時間 CI 圖

此圖同圖(9)，實驗次數少時，樣本數不夠，波動較為大，實驗次數多後，藉由信賴區間圖來觀察，最後慢慢趨向於穩定。

由圖(8)-圖(9)可觀察到藉由此文章所提出之整合模擬實驗方法來做實驗模擬，並不會造成實驗錯誤，可觀察到實驗數據是正常的。

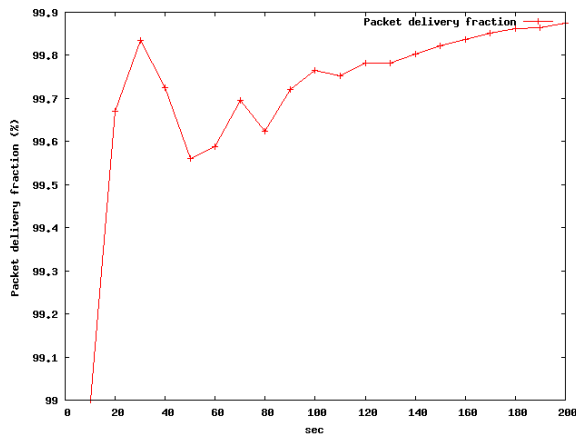
4.2.2. 針對單次模擬的時間分析

經過多次實驗觀察後，此小節針對單次實驗之各個時間點作分析，來觀察實驗結果做評估。



圖(11)：單次平均點對點延遲時間

由此圖可觀察到，一開始節點路由表尚未建構，傳送時需要花費較多時間，漸漸地路由表較為完整，決定路徑也越快。



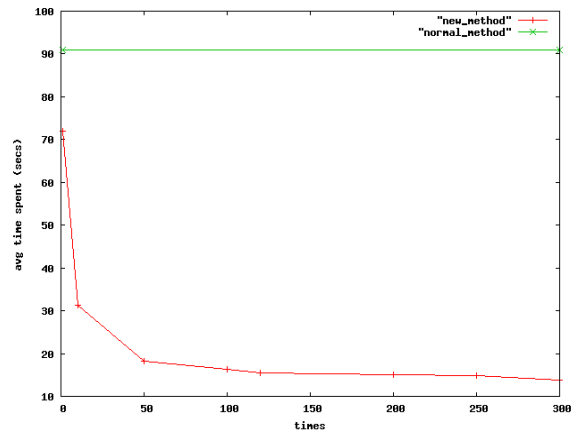
圖(12)：單次封包送達比例

在此圖中，無線網路封包傳送過程中，很容易遺失，且需要考慮重傳，在傳送過程越複雜時，越容易遺失。藉由圖(11)與圖(12)實驗圖表可觀察到實驗是合理化的。

5. 效能分析比較

為了清楚的了解一般的方法及本文改善後的方法在效率上的差別，此節做一個時間的統計，以證實文中提出的方法是有效率的。計算時間的方式，主要是針對於操作實驗及一開始的環境設定所花費的時間做記錄。最後，為了讓讀者更清楚時間上的差別，此處將時間除以次數做平均，以“平均花費時間/次”做比較，整理後的記錄如圖(13)。

圖(13)X座標為次數，Y座標為各個次數之下平均執行實驗的花費時間；上方的線為一般的做法，下方的線則為我們改善的方法。由圖(13)將可看出，隨著實驗次數的增加，改善後的方法之平均執行時間與一般方法之平均執行時間相比，明顯的減少了許多。這是因為原有的方法每執行一次都必須從頭執行每個相關程式、步驟，而改善後的方法只有在一開始需要設定微量參數，在每一次新的實驗中並不需要一再的重新執行每個步驟，而可直接等待實驗結果，所以實驗次數越多，越可以明顯看出花費時間的改善程度。



圖(13)：效能分析比較圖

此圖中，一般的實驗方法一直重複同樣的步驟，所以呈現直線的現象；而改善後的實驗方法，因為只在一開始需要設定參數、實驗次數，不需每次實驗都重新執行每個步驟，所以隨著實驗次數增加，平均執行時間反而越下降。

6. 結論

NS-2 是廣為使用的網路模擬器，擁有其重要的地位，但卻不是一個快速入門的工具，我們採用整合實驗步驟的方式改善使用上的便利性，藉此達到使用的方便性與快速性。此環境改進了原方法之缺點——設計實驗、計算數據及繪出圖表都需花上許多時間、許多步驟，且只能執行一次。但是經由整合模擬環境之開發，實驗將可快速執行多次，並且不需要一個一個執行所需軟體與各個指令。現在只需要一開始將參數設定好，即能等著結果出來，使用效率上大幅度增加。而使用者更可以藉著此方法簡易修改成自己實驗所需的環境來做實驗。

參考文獻

- [1] 林益宏、連韋荏、鄭彥杰、蔡其達，“設計 WiMAX MAC 層模擬模組 NS-2 模擬器當

- 紅” , http://www.2cm.com.tw/technologyshow_content.asp?sn=0708290018, 新通訊 2007 年 9 月號 79 期, [2008/12]
- [2]柯志亨、程榮祥、謝錫堃、黃文祥, ”計算機網路實驗－以 NS-2 模擬工具實作”, 2006/12。
- [3]柯志亨團隊編著, ”計算機網路實驗－使用 NS-2 模擬多媒體通訊與無線網路”, 2007/10。
- [4]The Network Simulator-ns-2, <http://www.isi.edu/nsnam/ns/> [2008/12]
- [5]NS2 教學手冊 (NS2 Learning Guide), <http://140.116.72.80/~smallko/ns2/ns2.htm> [2008/12]
- [6]NS by Example, <http://nile.wpi.edu/NS/> [2008/12]
- [7]Gao Fang, Lu Yuan, Zhang Qingshun, and Li Chunli. Simulation and Analysis for the Performance of the Mobile Ad Hoc Network Routing Protocols. The Eighth International Conference on Electronic Measurement and Instruments, 2007