利用可變速率方法賦予網路電話壅塞控制能力

中文摘要

近年來，一個具有壅塞控制機制的傳輸協議 DCCP 被提出，期能取代 UDP 成為不可靠傳輸的主流協議。我們以 NS-2 網路模擬器和實際網路進行實驗，發現 DCCP 無法與其他傳輸協議公平分享頻寬，因此現行 DCCP 的設計，尚無法完全取代 UDP。此外，目前 DCCP 以調整封包間隔的方式進行壅塞控制，也不適用於講求時效性的網路服務。

本研究首先以實驗證明，當網路情況不佳時，DCCP 無法與其他傳輸協議公平的分享頻寬；當使用 DCCP 傳輸越洋長距離網路電話，如遇頻寬不足時，會因頻寬競爭力較弱而無法維持通話品質。本研究提出可變速率方法(Flexible Bit-rate)調整時效性網路服務的封包大小來進行壅塞控制，在維持一定服務品質之前提下，促進網路的和諧。我們在一個實際網路的實驗環境中評估以UDP、DCCP及可變速率三種方式傳輸網路電話封包的效能，結果顯示透過可變速率方法，能有效降低網路電話的封包遺失率，維持通話品質。

Congestion Control Enabled VoIP by Flexible Bit-rate

Abstract

With congestion-control ability, Datagram Congestion Control Protocol (DCCP) is expected to replace UDP as a mainstream unreliable transport protocol. But our study found that DCCP is not able to get a fair share of bandwidth under the competition of others transport protocols no matter in NS-2 simulation or real world networking environments. Furthermore, any congestion control protocol that postpones the transmission of packets may not be adequate to support time-sensitive network services.

To maintain the quality of time-sensitive network services as well as to be TCP-friendly when facing network bandwidth fluctuation, we propose a Flexible Bit-rate congestion control mechanism for VoIP to adjust their data rate. Our experiments show that Flexible Bit-rate congestion control method could effectively reduce the packet loss rate and to maintain VoIP quality as compared with UDP and DCCP. Furthermore, it can have a much better bandwidth efficiency and adjust better to network fluctuation.

誌謝辭

誠摯的感謝連耀南教授在我研究生活中不辭辛勞的傳道、授業、解惑，這些知識都是我日後享用不盡的寶藏。

丁諭祺 October 2010

目錄

圖目錄

表目錄

第一章 緒論

現行的網路應用程式都希望能取得適當且公平的網路頻寬來完成它們的工作，另一方面，網路中的路由器等各個網路環節也盡力的避免造成網路壅塞的情形發生。傳輸層傳輸協議除了負責在傳送端和接收端之間傳輸資料之外，對於傳輸速率的控制也扮演了相當重要的角色，目前的網路應用程式多半使用 UDP[RFC 768]和 TCP[RFC 793]這兩個傳輸協議來傳遞資料。

UDP 中沒有確認封包、重送機制和連線逾時的概念，當資料被送出之後，發送端無法得知資料是否成功的被送達目的地；所以 UDP 無法測量傳送端與接收端之間的頻寬大小，更不可能藉由調整自身的發送速率來適應網路環境，因此 UDP 的傳輸速率多半是在傳輸之前先行設定，在整個資料的傳輸過程中不再改變，UDP 對於網路壅塞並無任何調節作用。

另一方面，TCP 由於加入確認封包、重送機制和連線逾時的概念，在資料傳輸的過程中，透過 ACK 封包可以偵測傳送端與接收端之間的網路情形，並透過壅塞控制機制，在傳輸的過程中改變傳輸速率，調節傳送端與接收端之間的網路負載。因此，TCP 承擔了網路壅塞控制的責任。

當整體網路中 TCP 的比例較大時，透過 TCP 的自我速率調整控制，網路的壅塞尚能獲得適度的控制，但隨著多媒體網路應用的興起，例如網路電話與視訊實況轉播，這類型的網路應用程式因必須維持恆定傳輸速率，大多使用 UDP 作為傳輸協議，因而網

路中將有越來越多的 UDP 封包，僅僅透過 TCP 的壅塞控制來做調節網路負載，將不足以維持整體網路的穩定。

因此 DCCP 這種具有壅塞控制機制的傳輸協議被提出，期望取代 UDP 成為不可靠傳輸的主流協議[10]。但與其他傳輸協議共存於網路上，是否能公平的分享頻寬，則不得而知；另外目前 DCCP 所使用的壅塞控制機制，也可能不適用於講求時效性的網路應用程式。

本研究將以實驗探討 DCCP 是否能公平的分享頻寬，並針對時效性的應用程式提出一套有效的壅塞控制機制，在保證一定的傳輸品質下，促進整體網路的和諧。

1.1 數據壅塞控制協定(DCCP)

多媒體應用是未來網路的趨勢，人們透過網路即時的傳送影音資訊，所以勢必設計一套講究即時性並且提供壅塞控制機制的傳輸協議，以避免壅塞崩潰的發生；以這個構想為前提，數據壅塞控制協議(Datagram Congestion Control Protocol, DCCP)在網際網路工程工作小組(Internet Engineering Task Force, IETF)努力之下就此誕生，在未來將取代 UDP 成為非可靠傳輸協議的標準。

DCCP 定義於[RFC 4340]，它是一個可以進行壅塞控制的不可靠傳輸協議，故沒有重送機制的設計，在 DCCP 在封包標頭中記錄著連續的編號，藉此偵測封包遺失的情形來達成壅塞控制，另外 DCCP 提供多種壅塞控制機制，在傳輸開始時可由使用者透過壅塞控制編號來做選擇。

1.2 DCCP-Based VoIP

語音通話對於聲音傳遞時間要求極為嚴苛，當聲音從發話者的口中傳送至接收者的耳中，必須維持在 400 毫秒以下，如果超過這個時間，便會嚴重影響到語音對話的互動性。因此 VoIP 對於封包傳遞的時間分秒必爭，且 VoIP 必須維持恆定傳輸速率，大多採

用 UDP 作為傳輸協議，但若將傳輸協議改為 DCCP，是否能維持原本所需的傳輸效能，並維持穩定的通話品質？我們必須先探討兩個問題。

第一個是不同於沒有壅塞控制機制的 UDP，加入壅塞控制機制的 DCCP 與其他傳輸協議共存於網路上，是否能公平的分享頻寬？

第二個是 DCCP 壅塞控制機制，是否適用於 VoIP？目前的 VoIP 在通話過程中，語音壓縮率是固定的，因此每秒所產生的資料量是固定的；但現行的 DCCP 壅塞控制機制，是透過調整封包發送間隔來降低傳送速率，當傳輸速率低於 VoIP 所需，VoIP 產生的封包會暫存於 buffer 之中，語音封包傳送至接收端的時間勢必加長，因此透過暫緩送出封包的壅塞控制機，可能無法適用在 VoIP 之上。

1.3 研究目的與方法

本研究將以實驗的方式，探討目前 DCCP 的壅塞控制機制，在 TCP 的頻寬競爭下，是否能公平的分享頻寬。

此外將提出一套具有壅塞控制機制的 VoIP 速率控制來促進網路的和諧，並維持一定的 VoIP 通話品質；以往的 VoIP 速率控制的研究只考慮到 VoIP 自身的品質，並未顧及整體網路的和諧。

不同於一般資料傳輸的壅塞控制機制，在 VoIP 上實作壅塞控制機制必須考量到封包延遲等問題。由於現行的 VoIP 的語音壓縮速率在通話過程中是不會改變的，透過改變語音封包送出的間隔時間來進行壅塞控制的方式，可能無法維持 VoIP 通話品質；因此我們提出一套偵測網路壅塞的方法，配合應用程式，在通話的過程中以改變語音壓縮率的方式來調整語音封包大小進行壅塞控制，舒緩網路壅塞的情形，並維持一定的通話品質。

1.4 章節安排

本論文共有三大主軸，其一探討 DCCP 與其他傳輸協議的頻寬競爭公平性，其二是現行 DCCP 的壅塞控制機制運用在即時性網路應用程式上的適切性，最後提出一套針對即時性網路應用程式的壅塞控制方法。

在二章中提到了壅塞控制機制的相關背景及文獻回顧，第三章中則以實驗的方式探討 DCCP 與其他傳輸協議的頻寬競爭公平性。第四章說明我們提出的調整封包大小的壅塞控制方法，並在第五章進行實驗與效能分析，最後於第六章進行結論。

第二章 背景與相關研究

2.1 壅塞崩潰

壅塞崩潰(Congestive Collapse)這個問題最早被發現於 1984 年，第一次觀察到這個現象是在 1986 年 10 月，當時 NFSnet 的骨幹網路的可用頻寬從 32Kbps 下降到只有 40bps[RFC 896]。其原因在於早期的 TCP 版本在壅塞控制缺乏良好的設計；當網路壅塞的情況發生時，TCP 於發送端將封包發送出去之後，沒辦法在重送逾時(Retransmission Timeout)的時間內收到接收端所回覆的確認封包(Acknowledgement Packet)，TCP 將視為一個封包遺失(Packet loss)事件而啟動重送機制；但發送端沒有在重送逾時時間內接收到來自接收端的確認封包，並非全都是因為封包遺失，而有可能是封包在傳送的過程中遭遇到網路壅塞的情況，使得封包延遲(Packet Delay)時間拉長，早期的 TCP 版本在這些情況下會持續重送封包，將使得網路壅塞情況越來越嚴重而造成壅塞崩潰，因此如何設計一個好的壅塞控制機制，一直是網路中相當重要的課題[1]。

2.2 壅塞控制

壅塞控制這個概念最早於 1986 年由 Nagle 在[13]中提出之後，網路中關於壅塞控制機制的研究就沒有停過，直到 1988 年 Jacobson 在[11]中詳細討論了慢啟動、快速重傳、壅塞避免等壅塞控制的方法，成為現今最常用的 AIMD(Additive Increase Multiplicative Decrease)壅塞控制機制的基礎，目前基於 AIMD 為基礎的壅塞控制方法有 TCP Tahoe、

TCP Reno、TCP NewReno、TCP SACK 等等，在[5]中透過模擬的方式，分析了這幾種壅塞控制機制的效能。

但隨著多媒體網路應用的興起，越來越多應用程式透過 UDP 傳輸資料，通常這些網路應用程式不需使用 TCP 的可靠傳輸的特性，例如 IPTV 和 VoIP 等等。但由於 UDP 缺乏壅塞控制機制，這些應用程式必須針對網路壅塞的情況自行設計應對方式，然而在應用層實作壅塞控機制並不容易，因此有一些針對不可靠傳輸進行壅塞控制的傳輸層傳輸協議被提出。

2.3 數據壅塞控制協定(DCCP)

DCCP 是個連線導向的傳輸層傳輸協議，包含了連線的建立與拆除、ECN、壅塞控制等，它提供了不同的壅塞控制方式供使用者選擇，但並不包含重送機制，是一個不可靠的傳輸協議[10]。DCCP 和 TCP 相當類似，透過 ACK 封包來確認資料是否送達接收端，ACK 訊息中包含了發送端送出的封包是否到達目的地，以及標頭中是否含有 ECN 標記。在連線建立的時候，使用者可透過壅塞控制編號(CCID)選擇不同的壅塞控制機。

2.4 TCP 的壅塞控制

由於網路並非完美無缺，少部分的封包可能在傳遞的過程中遺失。一般而言，網路壅塞是造成封包遺失的最大原因，此時傳輸協議必須要盡快做出反應降低傳送速率以舒緩網路的壅塞。TCP 使用 AIMD 的方式調整傳輸速率，此外還有慢啟動、壅塞迴避等狀態來避免網路壅塞[11]，以上這些方式最早出現在 TCP Tahoe 和 Reno 版本中[5]。

2.4.1 TCP Tahoe 和 Reno

在每條 TCP 建立的連線之中，發送端利用壅塞視窗之控制來限制兩個終端節點之間未經確認的封包總數上限，亦即已送出但尚未到達的封包總數之上限，藉此來控制傳輸速率。當接收端收到一個封包，便會回覆一個 ACK 給發送端，告知目前所收到最新

的封包序號；當發送端收到 ACK 之後，便再送出一個封包，壅塞控制窗口內的未被確認封包總數維持於窗口上限。若接收端收到跳序的封包時，它會送回一個 duplicate ACK 重複前一個封包序號給傳送端，藉以回報異常狀況。

TCP 在連結建立的開始或是逾時產生後會進入慢啟動機制[16]。接收端每收到一個封包，便回送一個 ACK，而傳送端在接收到每一個 ACK 後便增加壅塞視窗之大小，以倍數的方式遞增直到產生封包遺失，當接收端發生連線逾時，CWND (Congestion Window)會降至初始值，再以倍數成長至 threshold，此時便進入壅塞避免階段，此因既已偵測到傳輸速率之極限，理當調整減緩 CWND 的擴張速度。當 CWND 增長至 threshold 之後每接到一個 ACK，CWND 只增加一格，這個過程持續不斷的進行，直到壅塞產生後，縮減 CWND 並將 threshold 減半。而 TCP Tahoe 和 Reno 對於收到 duplicate ACK 有著不同的處理方式。

當傳送端收到三個 duplicate ACK 時，TCP Tahoe 處理的方式和處理連線逾時相同(即等到 Timeout 才啟動封包遺失相關動作)，而 Reno 則認為等到 Timeout 發生後再進行重傳常會耗時過久，影響效能，因此加上了 Fast Retransmit 機制，當傳送端收到三個 duplicate ACK 後，便假設此重複 ACK 所回報序號之下一個封包已經遺失，但網路只有輕微壅塞故立即重傳該封包，並進入 Fast Recovery，不須等到 Timeout。

在 Fast Retransmit 之後，所進入的狀態是擁塞避免階段，而非慢啟動階段，此機制稱為 Fast Recovery，之後如果等到連線逾時仍然沒有收到 ACK，TCP Reno 將會視為網路嚴重壅塞，並進入慢啟動狀態。

2.4.2 TCP Vegas

TCP Vegas 利用封包來回時間(Round-Trip Time)來調整 CWND[2]，並使用遞增的方式來增加 CWND 的大小，雖然這種測量網路頻寬的方式較為準確，但相較於其他 TCP 版本，TCP Vegas 對於頻寬的競爭力較為薄弱，所以不受一般人青睞。

2.4.3 其他版本 TCP

NewReno 和 SACK 的出現是用來改善並增進 Reno 的效能[5, 6]。在特殊的網路環境中，例如無線網路，封包遺失的主因不再是網路壅塞，原本的壅塞控制機制反而會造成傳輸速度緩慢，因此有許多不同的 TCP 版本被提出法來改善這些問題[9, 14]。

2.5 DCCP 的壅塞控制

DCCP 目前定義了 CCID2 與 CCID3 兩種不同的壅塞控制方法。

2.5.1 CCID 2: TCP-Like 壅塞控制

在 DCCP 中 CCID2[7]制定的壅塞控制機制是和 TCP 相當類似的，其中也有幾個重要的狀態，分別是慢啟動(Slow Start)、壅塞迴避(Congestion Avoidance)和重送逾時(Retransmission Timeout)，最大的不同在於沒有重送(Retransmission)機制。如同 TCP 一般，在 CCID2 中利用壅塞視窗(Congestion Windows)配合 AIMD 的演算法來控制傳輸速率，另外 CCID2 也針對 ACK 封包進行壅塞控制。

2.5.2 CCID 3: TFRC 壅塞控制

TFRC 是以傳輸速率(Rate-Based) [8]為基礎的壅塞控制機制。不同於 TCP 使用 CWND，TFRC 而是以(1)所列計算式來估計並控制發送端的傳輸速率，主要透過 RTT(Round Trip Time)與封包遺失(Packet Loss)作為依據，並以計算出的 X 值為限制，控制傳輸速率。

$$X = \frac{s}{RTT * \sqrt{\frac{2*b*p}{3}} + (t\_RTO * (3 * \sqrt{\frac{3*b*p}{8}} * p * (1 + 32 * p^2)))} \quad (1)$$

s=平均封包大小
p=封包遺失率
RTT=封包延遲
t_RTO=TCP Retransmission Timeout 時間

TFRC 的壅塞控制方式必須在傳送端和接收端之間建立起連線，用以進行網路參數的回饋。當傳輸開始時，TFRC 效法 TCP 的慢啟動階段，以便快速的增加傳輸速率來取得公平的頻寬，等到發生第一個封包遺失事件後，傳送端會收到接收端傳回的回饋訊息(feedback)，進而結束慢啟動階段，並將接收端回饋的網路參數代入公式(1)中，重新預估可用的傳輸速率後，再以此傳輸速率發送數據封包。

在 TFRC 傳輸速率公式中(1)，最重要的兩個參數分別是封包遺失率與 RTT，這兩項參數的改變將會大幅影響傳輸速率的估計。TFRC 透過封包標頭中的序號，接收端可以計算出封包遺失率 p 回饋給傳送端；在由標頭中的時間戳記(timestamp)就可計算出封包往返時間 RTT。TFRC 假設在同一段時間內，造成封包遺失的原因都是相同的，故將一個 RTT 之內所有的封包遺失都視為一個封包遺失事件，因此估計出的傳輸速率會較為平緩。

2.6 VoIP 語音封包產生流程

VoIP 使用網際網路來傳送語音資料，其作法是將語音從麥克風擷取後，由類比訊號轉為數位訊號，經由壓縮演算法封裝為數據封包，透過網路傳輸，接收端後將數據封包解壓縮後，再轉換成為類比訊號由喇叭播出，如圖所示。

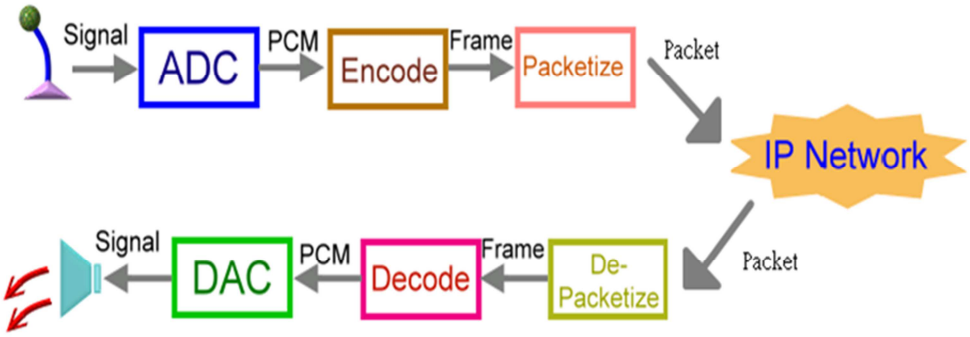

圖 1　VoIP 語音封包產生流程

2.6.1 取樣與編碼

麥克風將聲音轉換成為有強弱幅度變化的電壓信號，此信號是一種類比信號，經由數位轉換的過程後成為數位信號，才能由電腦處理。

在取樣過後，電腦將數位訊號進行編碼，在過程中通常會加入壓縮的動作來減少網路頻寬的消耗，在壓縮前必須先從取樣後的數位音訊訊號取出一小段，經由編碼器(Codec)編碼後輸出壓縮過後的資訊，每一段資訊稱為一個訊框(Frame)，一個訊框會包含 10 毫秒到 30 毫秒的音訊訊號，依不同的編碼器會有不同的標準。

在編碼的過程中，可以選擇不同的壓縮演算法。高壓縮率所產生的語音封包較小，音質較差，所需的網路頻寬也較小，相反的低壓縮率所產生的語音封包較大，音質較佳，但所需的頻寬也較大。目前針對語音壓縮演算法主要的有：G.723、G.729a、iLBC、Speex[19, 20, 21, 23]等，表 1 是幾個常見編碼器的參數。

表 1　各種常見語音壓縮編碼的參數

Codec	Sampling Rate	Frame Size	Bit-rate
G.723	8kHz	67.5ms	6.4Kbps
G.729	8kHz	20ms	8Kbps
iLBC	8kHz	30ms/20ms	13.3/15Kbps
Speex	8kHz/16kHz	20ms/40ms	2.15-44Kbps

2.6.2 封包封裝與傳輸

編碼之後透過網路傳輸前，會將訊框封裝(Packetization)成為封包後才交由網路傳遞，一般網路電話應用程式會將編碼過後的資料封裝成 RTP 封包，再加上傳輸層傳輸協議封包標頭與 IP 標頭後成為一個 IP 封包，當一段語音資料封裝成為 IP 封包後，由 IP 封包標頭中所記錄的位置資訊，即能透過網路傳送至目的地。

2.6.3 影響通話品質的參數

影響通話品質的參數可由使用者端和網路端兩個方向來說明。以使用者端來說，較差的收音設備和環境中的噪音，或是由喇叭發出聲音後再由麥克風接收產生的回音，甚至是不同編碼器(Codec)有不同的失真程度，都會降影響網路電話的通話品質；另外在網路傳輸的過程中，網路情況好壞會造成封包延遲(Packet Delay)、封包遺失(Packet Loss)、封包抖動(Jitter)等情形，諸如此類問題皆可能降低網路電話的通話品質。

2.7 VoIP 通話品質評量指標

網路語音品質評估指標分為主觀與客觀兩種。主觀的語音評估方式是利用人的耳朵直接收聽語音，並給此段語音評分，常見的指標為 MOS；客觀的方式是利用數學模型，將系統中的參數代入計算出分數，常見的指標有 E-Model。

2.7.1 MOS

一般常用來衡量 VoIP 通話品質的指標是依據 ITUT 所制定 MOS(Mean Opinion Score)分數[22]，MOS 是一種較為主觀的語音品質測量指標，收聽者按照從 1 到 5 分對語音會話品質來進行分級，分數由高至低分為最好和最壞，如表 2。

表 2　Mean Opinion Score 定義

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

2.7.2 E-Model

除此之外，還可以利用同為 ITUT 所制定的 E-Model[17]，E-Model 透過語音傳輸過程中若干因素，計算出對語音品質的負面影響綜，來評估該通話之品質水準。E-Model 中的 R-factor 中包含了所有影響通話品質的要素，最差為 0 分，最好為 100 分。

$$R = 94.2 - I_d - I_e \tag{2}$$

在公式(2)中I_d代表了 network delay、codec-related delay 和 playout buffer delay，I_e代表了語音壓縮所造成的語音品質下降和語音透過播放設備放出所造成的失真，另外在 E-Model 和 MOS 之間可以透過公式(3)進行轉換，表 3 列出 R-Factor 與 MOS 之間的分數對應。

$$MOS = 1 + 0.035R + 7 \times 10^{-6}R(R-60)(100-R) \tag{3}$$

表 3　R-factors 與 MOS 之對應

R-factor	Quality	MOS
90<R<100	Excellent	4.34 ~ 4.5
80<R<90	Good	4.03 ~ 4.34
70<R<80	Fair	3.60 ~ 4.03
60<R<70	Poor	3.10 ~ 3.60
50<R<60	Bad	2.58 ~ 3.10

2.7 現有改變語音壓縮速率的 VoIP 壅塞控制機制

以往的VoIP速率控制的研究只考慮到VoIP自身的品質，並未顧及整體網路的和諧；近年來因為網路電話的興起，VoIP 的壅塞控制才慢慢被重視。

在 Fariza Sabrina 和 Jean-Marc Valin 提出了一套具有壅塞控制機制的 VoIP 網路電話[15]架構，當中採用了 XCP(eXplicit Control Protocol)[4]傳輸協議來計算傳送端與接收端之間的網路頻寬，在圖 2 中畫出了此系統的架構圖。當 VoIP 通話建立時，此系統在發話方與受話方之間的通信閘(gateway)建立起 XCP 連線估計出可用頻寬，通信閘首先將使用者傳來的語音串流依照 XCP 所回報的頻寬，採用相對應的語音壓縮速率進行壓縮傳送至受話方的通信閘；當受話方的通信閘收到壓縮過後的語音串流後，再解壓縮傳給受話者。此系統透過通信閘控制語音壓縮速率調節進入網路中的資料量，進而達到壅塞控制的目的。

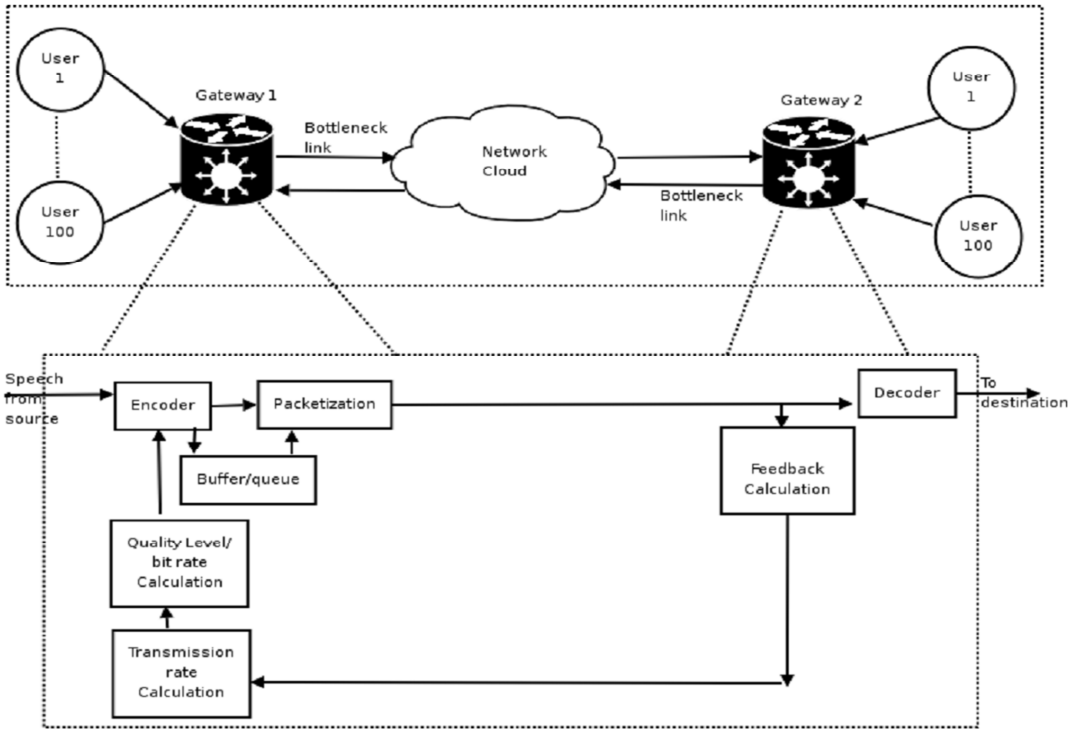

圖 2　Fariza Sabrina 等人提出具有壅塞控制機制的 VoIP 系統架構

這種 VoIP 架構中的優點在於每位發話者僅需將語音封包傳送給通訊閘，其餘的語音壓縮、解壓縮等動作都交由通訊閘處理；付出的代價為此通訊閘須要有足夠的運算效能與網路頻寬，且這種方式只能對通訊閘和通訊閘之間的網路鏈結進行壅塞控制。

第三章 數據壅塞控制協定頻寬競爭分析

我們使用 NS-2 網路模擬器[18]並裝上 Mattsson 所提供的 DCCP 模組[12]來進行實驗，評估 DCCP 在頻寬競爭中的表現，並測試是否能維持網路電話的通話品質。

3.1 實驗設計

圖 3 為實驗中所使用的網路拓樸，在中間的鏈結設置 0.25Mbps 的頻寬與 50 毫秒的延遲模擬越洋的網路鏈結，用以作為網路的瓶頸點，其他的鏈結間設置 10Mbps 的頻寬與 10 毫秒的網路延遲。

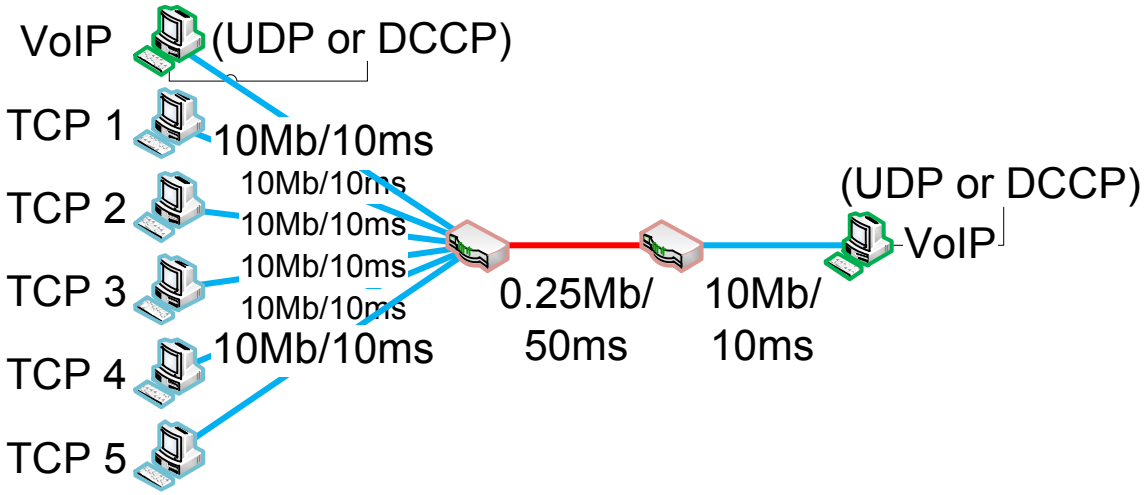

圖 3　越洋網路電話實驗拓樸

實驗中將以 UDP 和 DCCP(CCID3)作為比較。在第一個情境中，我們使用 UDP 或 DCCP 傳輸一通網路電話，每隔 10 秒鐘增加 1 條 TCP 資料流進入網路，直到網路中有 5 條 TCP 資料流；在第二個情境中，實驗開始時網路中就存在 5 條 TCP 資料流，網路電話在實驗的第 20 秒進入直到第 80 秒結束。這個實驗將使用五種不同版本的 TCP，分

別是 Tahoe、Reno、NewReno、SACK 以及 Vegas。表 4 中列出所有的實驗參數，表 5 是在 DCCP CCID3 中用來估計 TCP 吞吐率的方程式。

表 4　越洋網路電話之實驗參數

Parameters	Value
VoIP Packet Size	40 Bytes (Payload only)
VoIP Inter-Packet Interval	20 ms
TCP Versions	Tahoe, Reno, NewReno, SACK, Vegas
TCP Packet Size	1460 Bytes
Router Buffer Size	20 Packets
Buffer Management Scheme	DropTail (FIFO)
Link Bandwidth	0.25~10 Mbps
Number of VoIP session	1 (Full Duplex)
Number of TCP session	5

表 5　DCCP CCID3 參數

	Parameters	Initial_Value
s	Packet Size	1460 Bytes
R	Round Trip Time	3 sec
P	Loss Event Rate	0-1.0
t_RTO	TCP Retransmission Timeout Value	3 sec
b	# of Packets Acknowledged by a Single TCP ACK	1

3.2 頻寬競爭分析

本章節的討論重點在於吞吐率(throughput ratio)的比較。我們分別在傳送端和接收端記錄下傳送和接收到的網路電話和 TCP 資料流，並於紀錄接收端每秒所收到的資料量來計算吞吐率。在本實驗中我們將延遲時間超過 350 毫秒的網路電話封包，視為封包遺失，不列入吞吐率的計算。

實驗結果發現，UDP 就如我們預期的，在面對各種 TCP 本版的競爭時，都能維持穩定的吞吐率，在圖 4,6,8,10,12 裡顯示了 UDP 與各版本的 TCP 的吞吐率變化(TCP 1 是第一條 TCP 資料流)，隨著越來越多的 TCP 資料流進入網路中，UDP 依然能維持其吞吐率，但 TCP 的吞吐率下降，這表示 TCP 會調整自身的傳輸速率，以適應網路壅塞，但 UDP 並沒有這種機制。

另外在圖 5,7,9,11,13 中顯示，使用 UDP 傳輸的網路電話，在網路中存在數條 TCP 資料流的情況下，能競爭到所需頻寬，維持通話品質的穩定。透過 TCP 1 和整體 TCP 的吞吐率變化也能很明顯的發現，在 UDP 進入網路中後，整體的 TCP 吞吐率下降，當 UDP 離開網路後，整體的 TCP 吞吐率上升，這顯示 TCP 會調整自身的傳輸速率，以適應網路的情況。

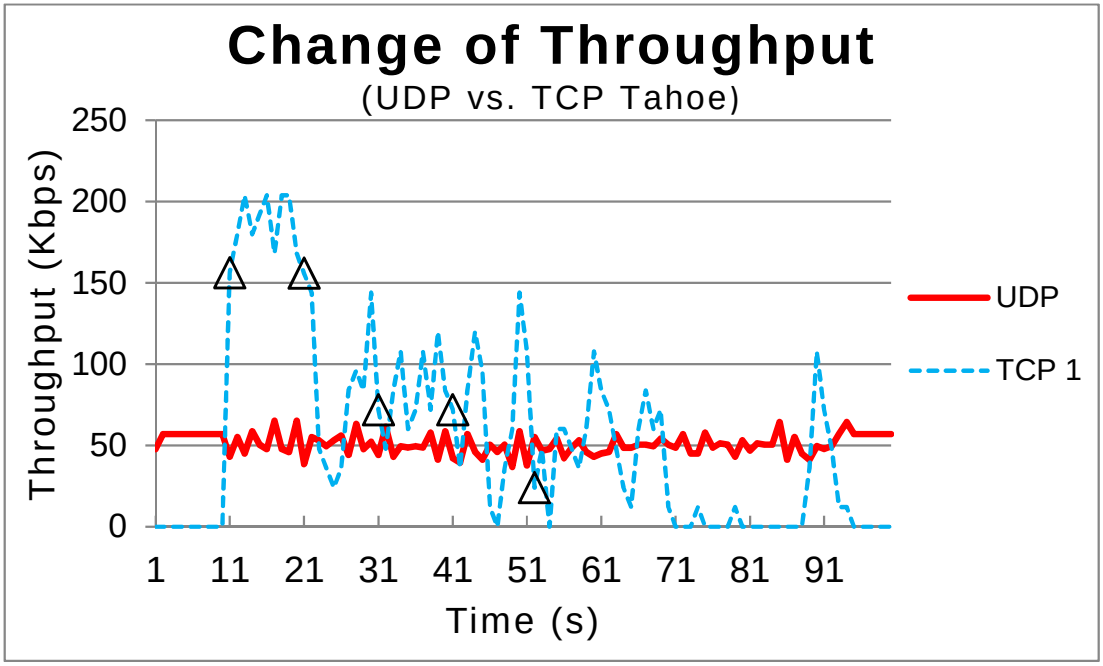

圖 4　UDP 與 TCP Tahoe 吞吐率比較-UDP 先進入

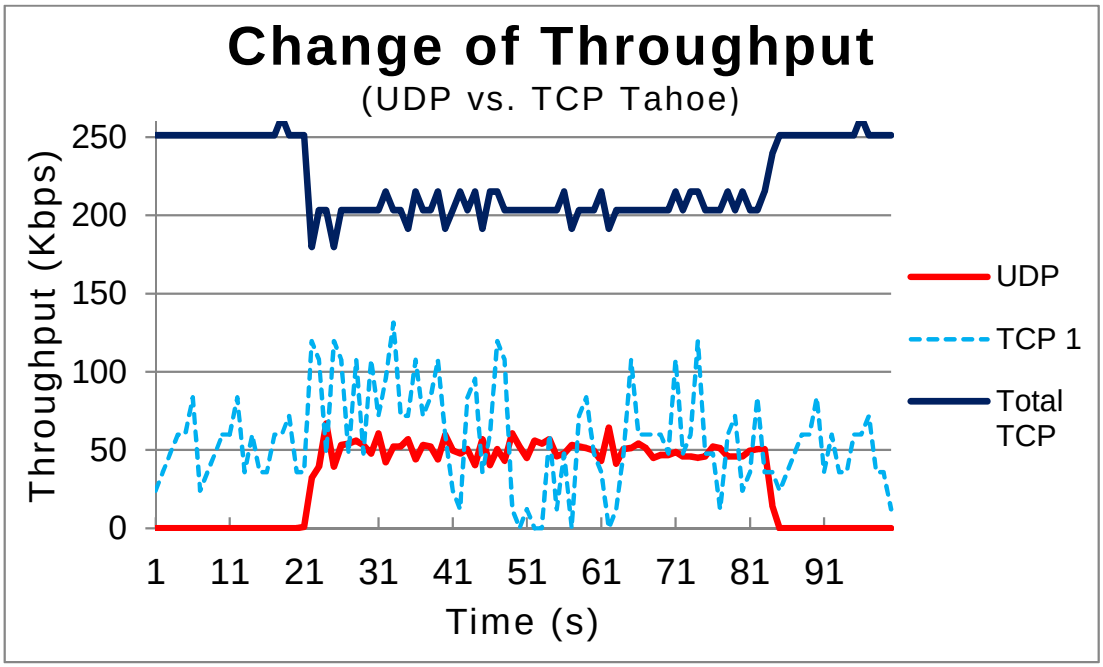

圖 5　UDP 與 TCP Tahoe 吞吐率比較-TCP Tahoe 先進入

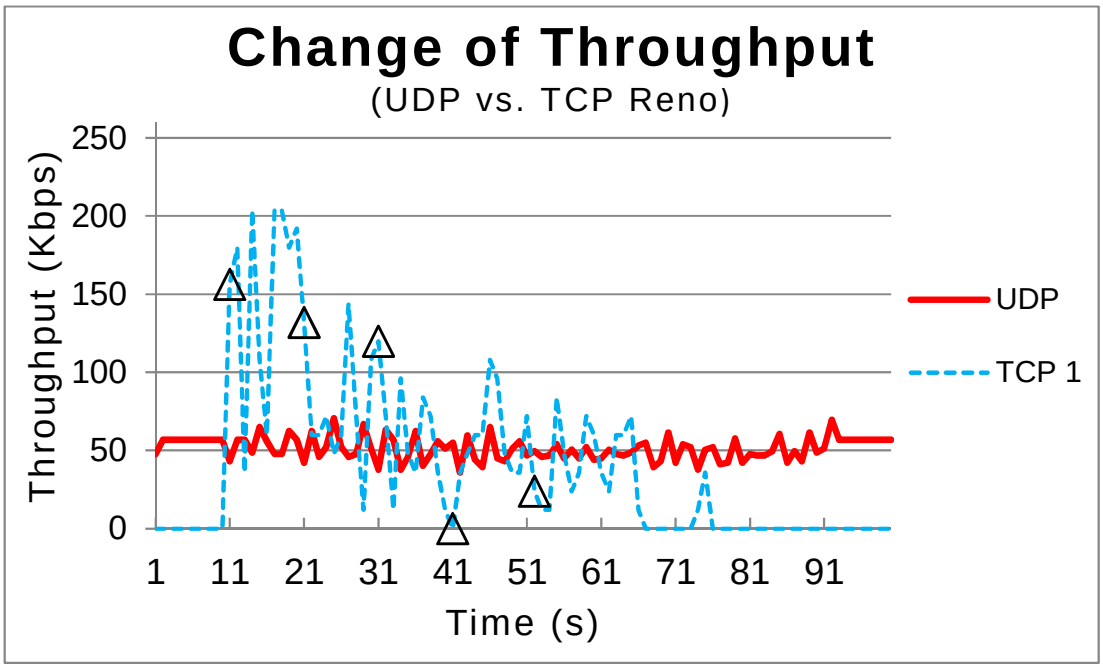

圖 6　UDP 與 TCP Reno 吞吐率比較-UDP 先進入

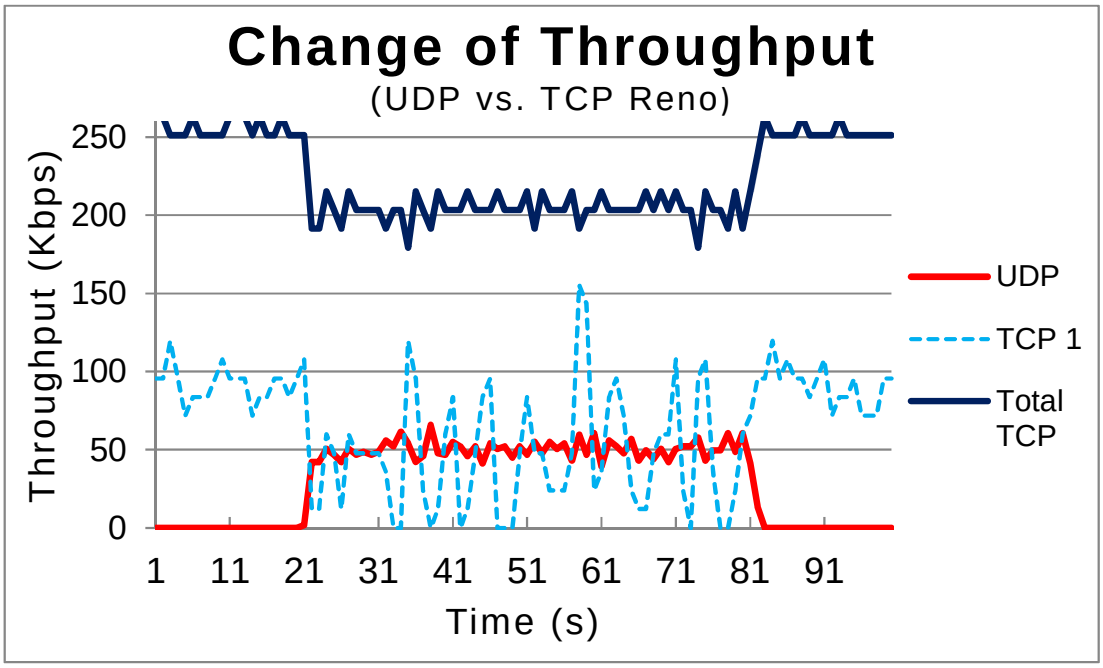

圖 7　UDP 與 TCP Reno 吞吐率比較-TCP Reno 先進入

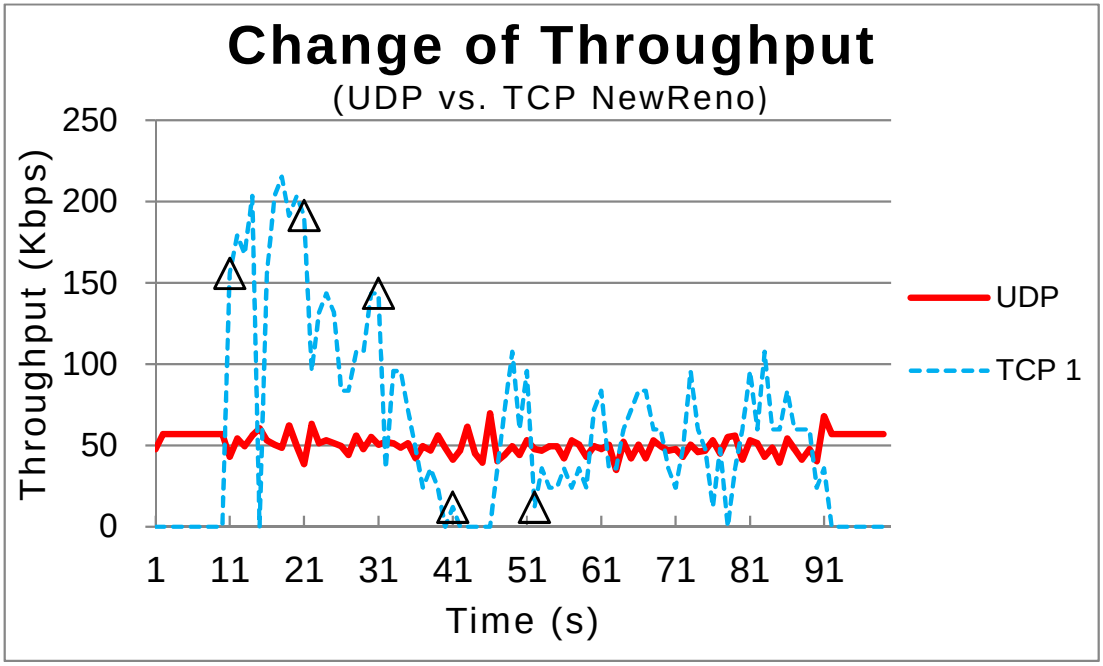

圖 8　UDP 與 TCP NewReno 吞吐率比較-UDP 先進入

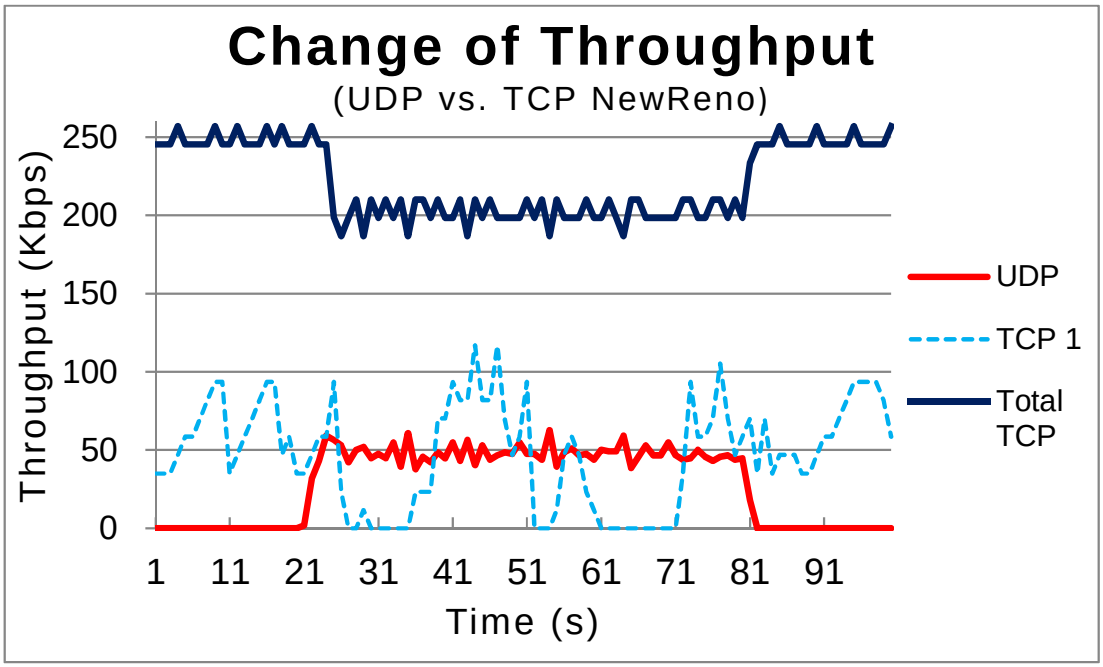

圖 9　UDP 與 TCP NewReno 吞吐率比較-TCP NewReno 先進入

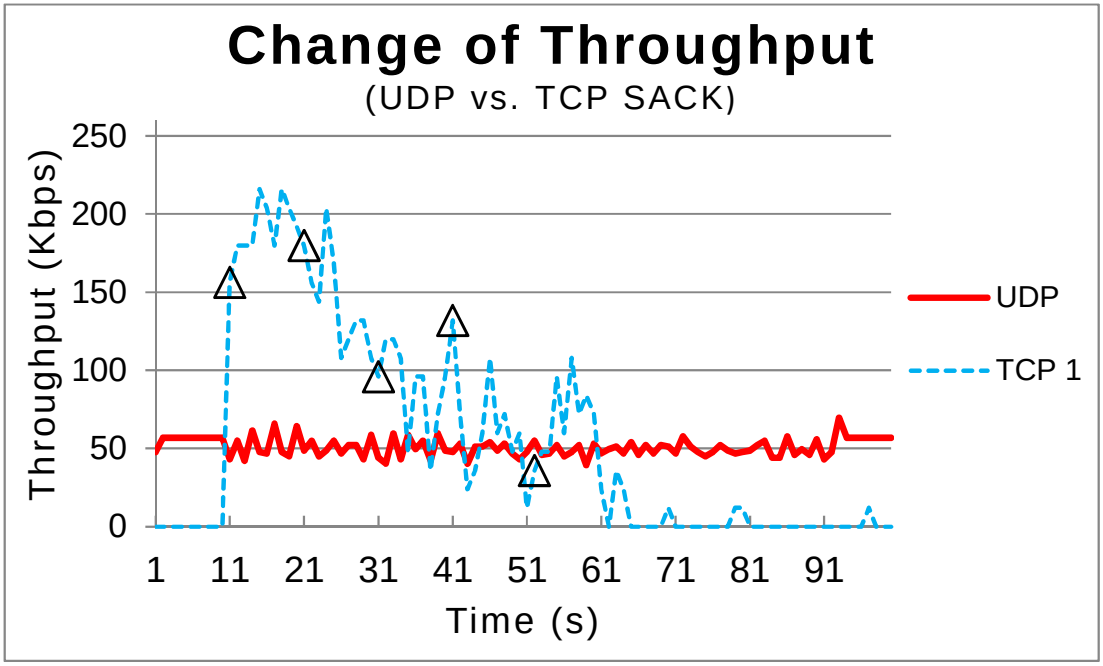

圖 10　UDP 與 TCP SACK 吞吐率比較-UDP 先進入

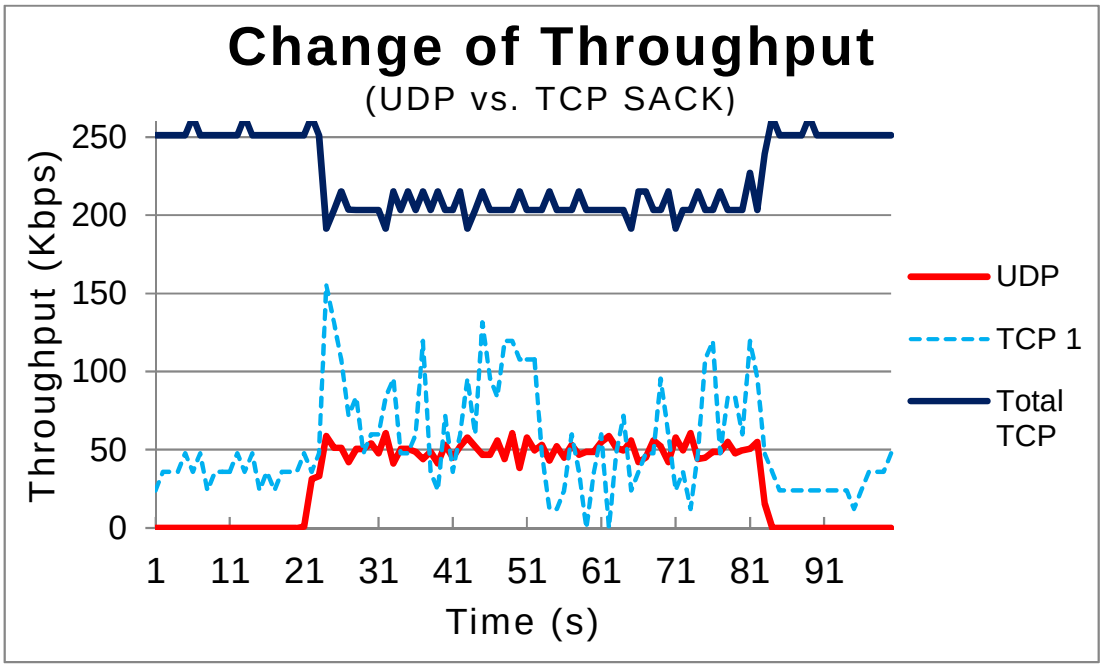

圖 11　UDP 與 TCP SACK 吞吐率比較-TCP SACK 先進入

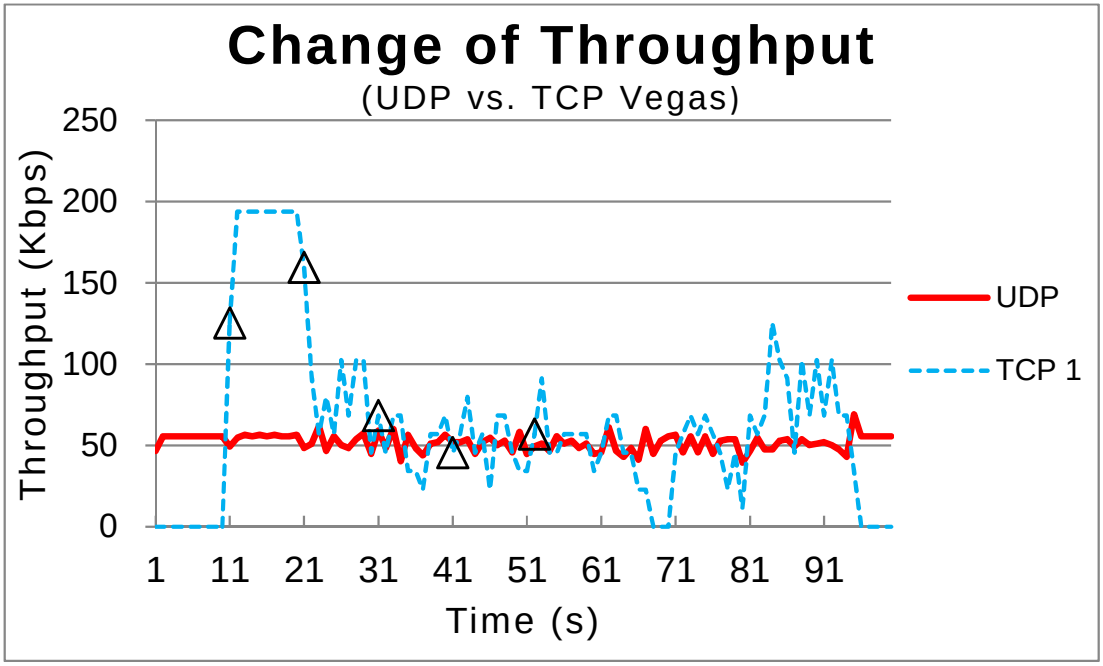

圖 12　UDP 與 TCP Vegas 吞吐率比較-UDP 先進入

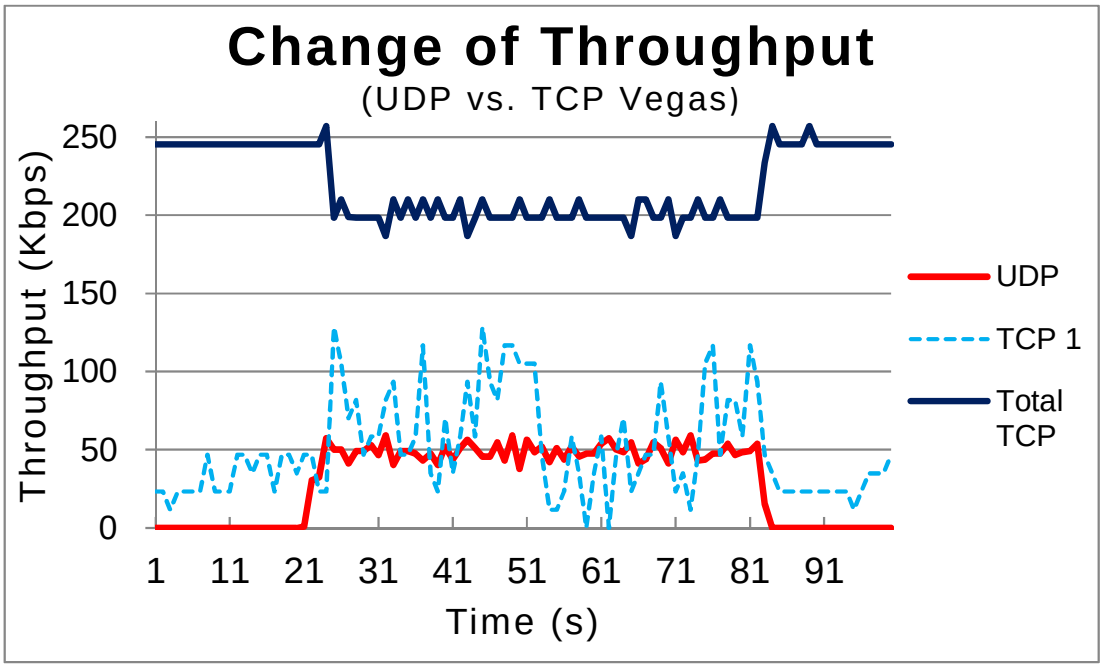

圖 13　UDP 與 TCP Vegas 吞吐率比較-TCP Vegas 先進入

在圖 14,16,18,20 裡顯示了 DCCP 與各版本的 TCP 1 的吞吐率變化，隨著後續 TCP 資料流的進入，DCCP 和 TCP 1 都受到影響，顯示出 TCP 與 DCCP 都自動調整了傳輸速率，以適應網路壅塞。但我們可以從圖中發現，DCCP 所爭取到的頻寬，遠小於各版本 TCP，甚至在只有一條 TCP 資料流的競爭下，DCCP 所爭取到的頻寬也只有初始值的 30%，這表示 DCCP 在網路中存在多條 TCP 資料流的情況下，無法公平的分享頻寬。

另外在圖 15,17,19,21 中顯示，使用 DCCP 傳輸的網路電話，在網路中存在數條 TCP 資料流的情況下，無法競爭到所需頻寬來維持穩定的通話品質。透過 TCP 1 和整體 TCP 的吞吐率變化也能很明顯的發現，在 DCCP 進入網路中後，TCP 吞吐率並無明顯的下降，這顯示 DCCP 的頻寬競爭能力遠小於 TCP，原因在於當網路壅塞時，封包遺失率和封包延遲增加，造成 DCCP 計算出的傳輸速率相當低。

另一方面，同樣以傳輸速率來作用壅塞控制的 TCP Vegas 對於頻寬的侵略性較其他 TCP 版本來得低，DCCP 所受到的影響較小。在圖 22 和圖 23 中顯示，五條 Vegas 資料流進入網路後，DCCP 所受的影響越來越大，在只有兩條 Vegas 進入網路時，利用 DCCP 傳送封包的 VoIP 沒有受到嚴重影響，但 VoIP 必須調整其傳輸速率(語音封包大小)來維持通話品質。

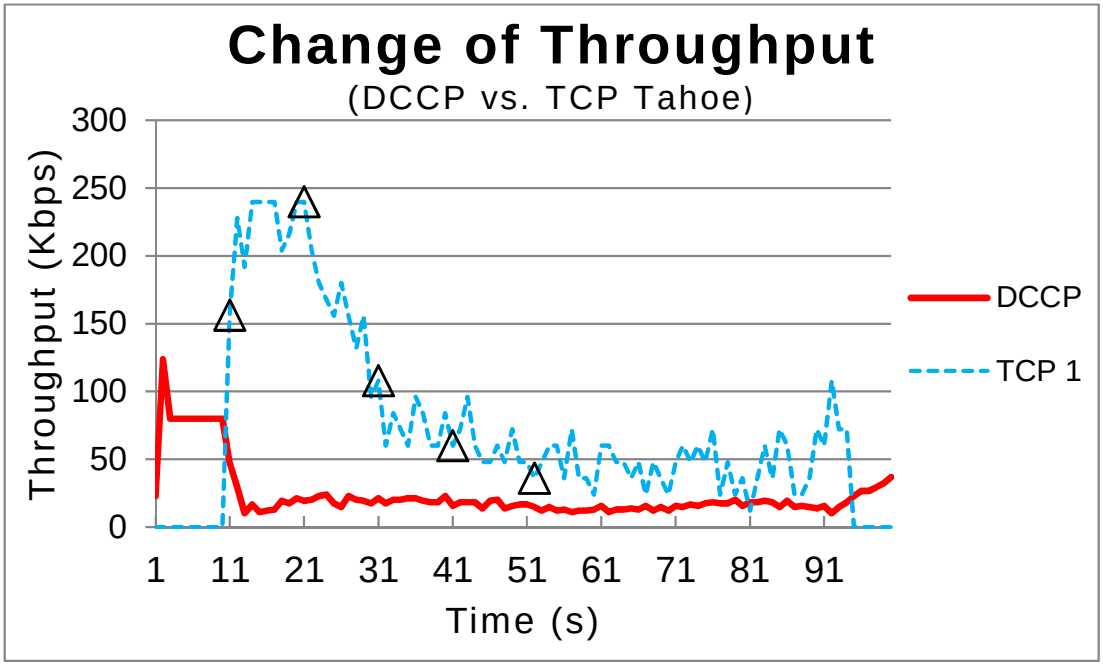

圖 14　DCCP 與 TCP Tahoe 吞吐率比較-DCCP 先進入

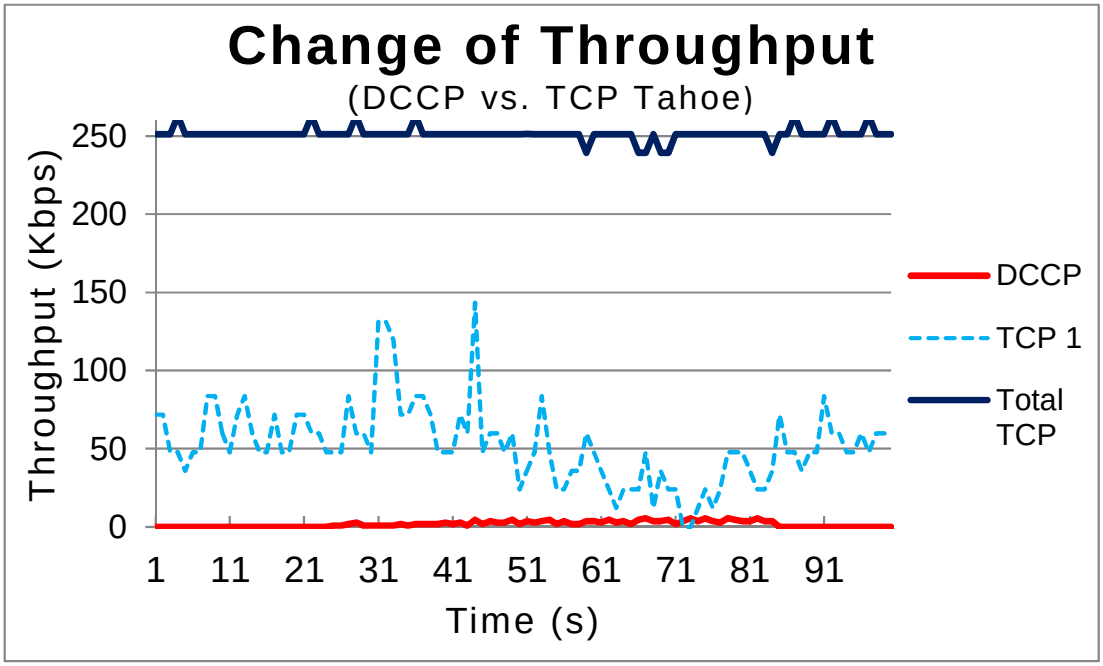

圖 15　DCCP 與 TCP Tahoe 吞吐率比較-TCP Tahoe 先進入

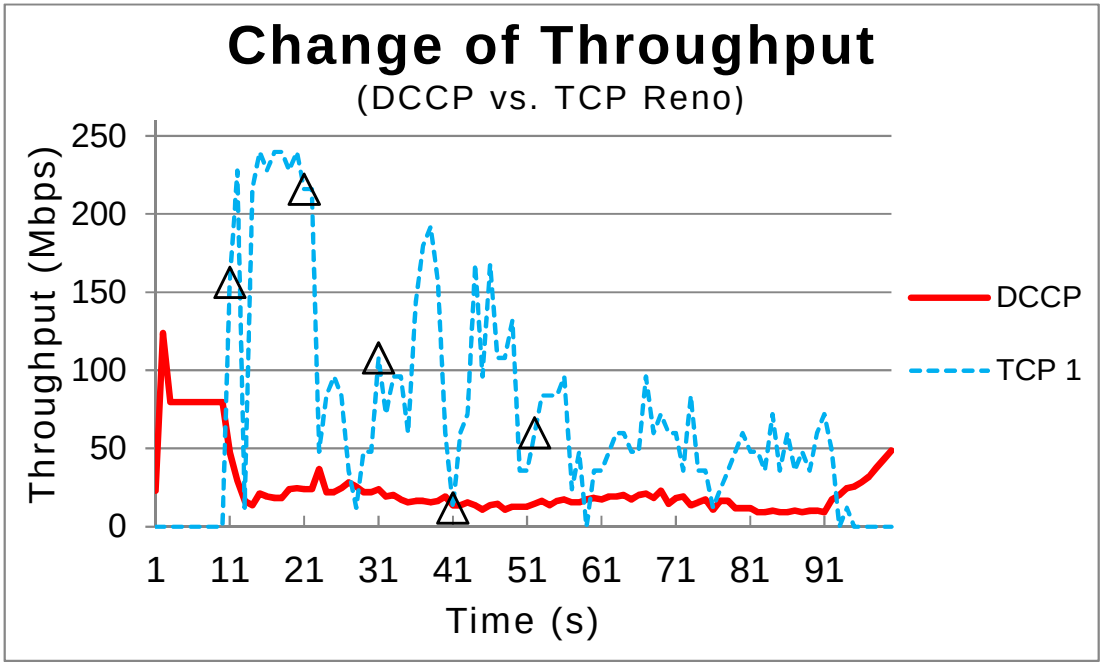

圖 16　DCCP 與 TCP Reno 吞吐率比較-DCCP 先進入

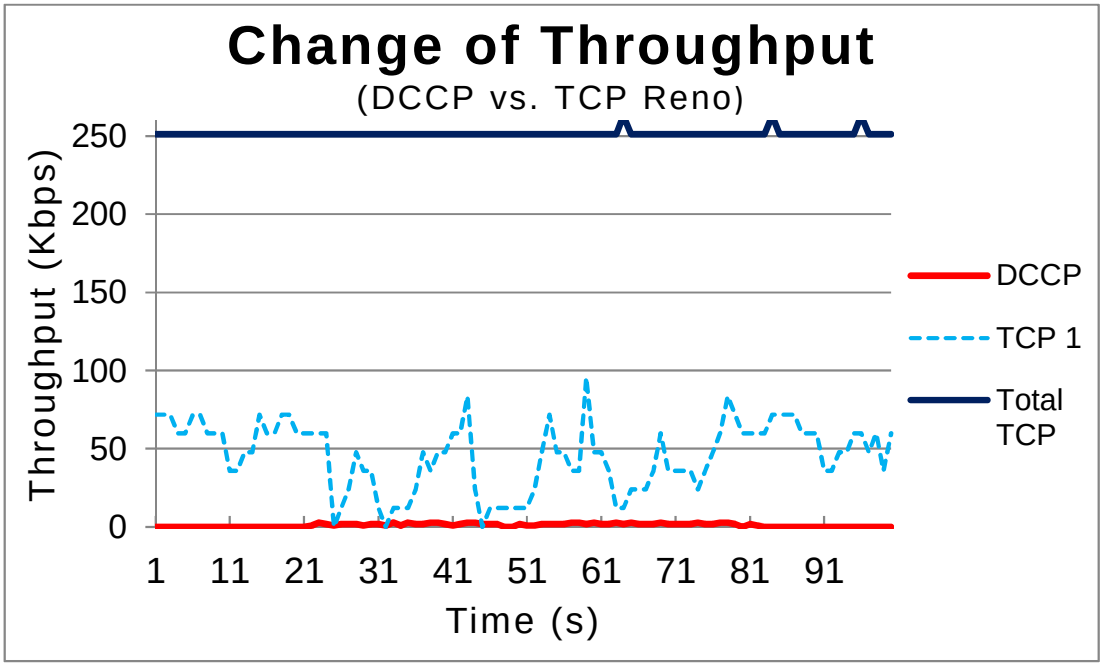

圖 17　DCCP 與 TCP Reno 吞吐率比較-TCP Reno 先進入

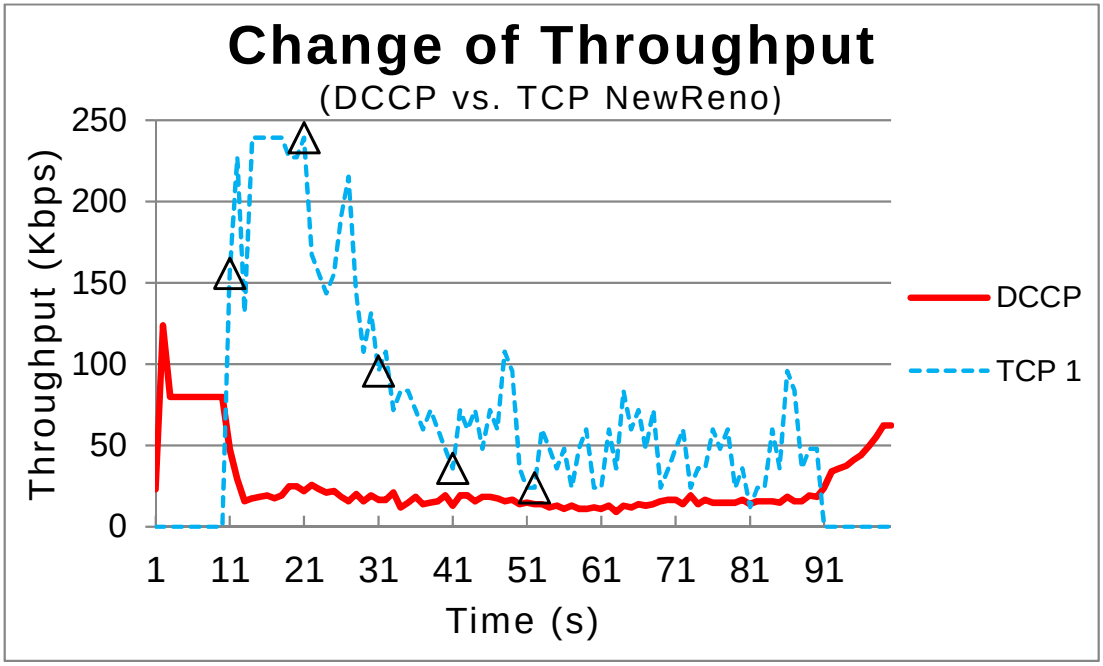

圖 18　DCCP 與 TCP NewReno 吞吐率比較-DCCP 先進入

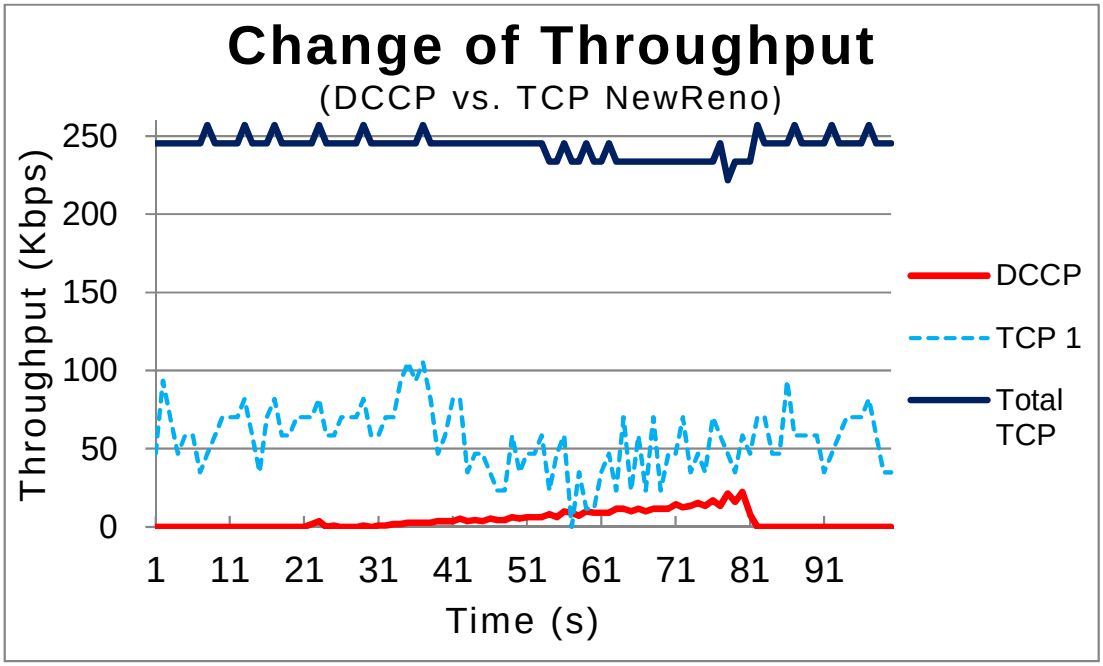

圖 19　DCCP 與 TCP NewReno 吞吐率比較-TCP NewReno 先進入

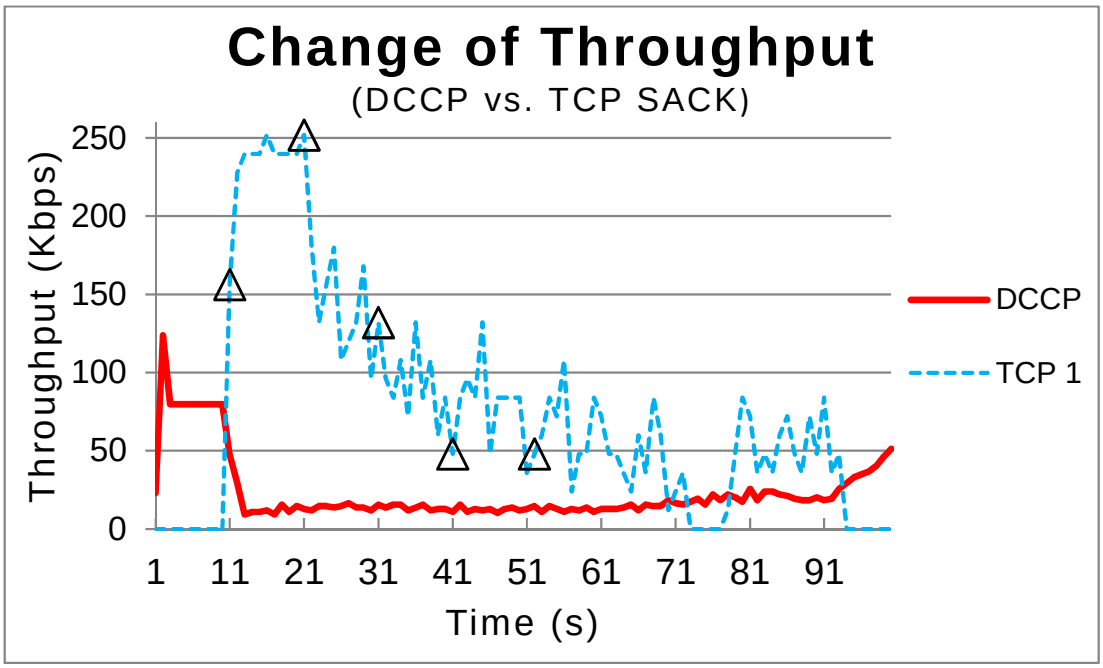

圖 20　DCCP 與 TCP SACK 較-DCCP 先進入

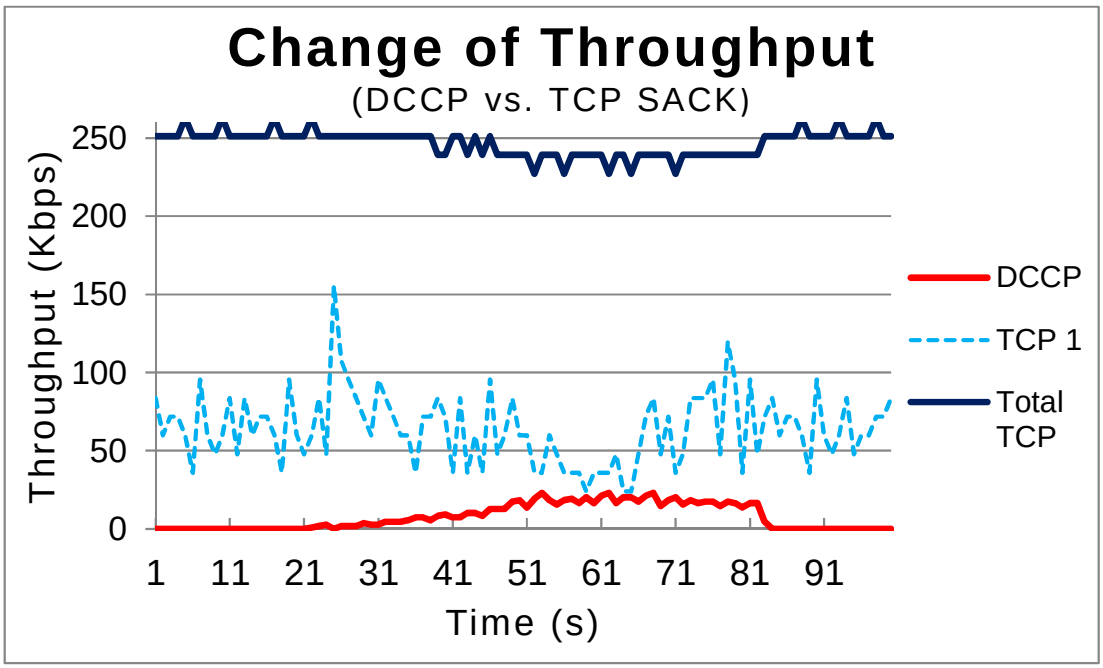

圖 21　DCCP 與 TCP SACK 較-TCP SACK 先進入

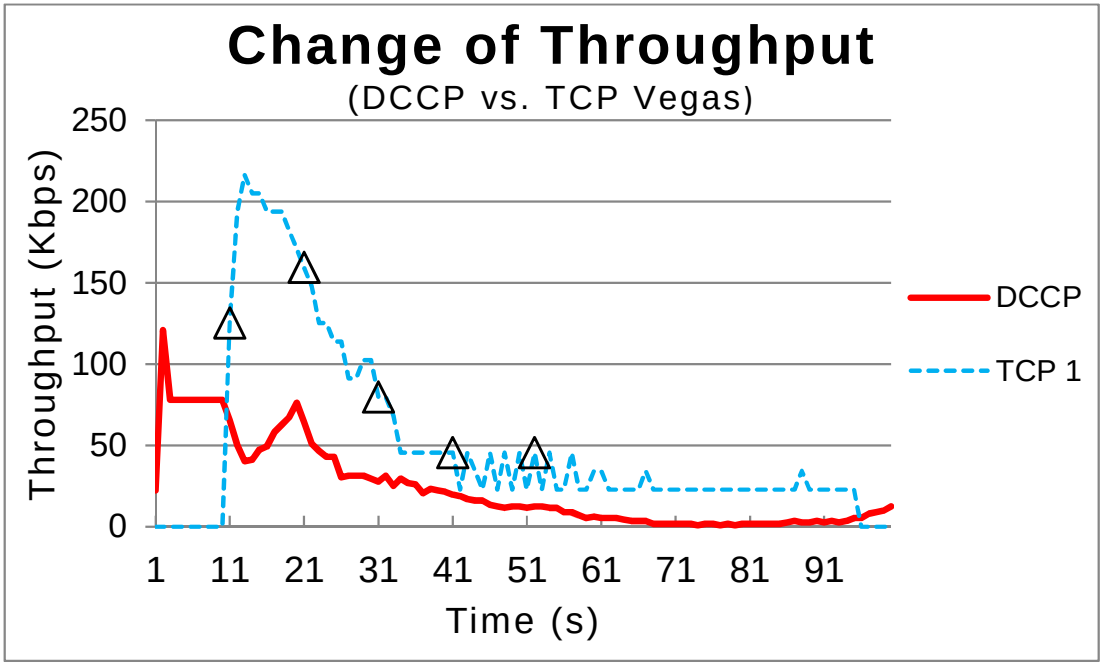

圖 22　DCCP 與 TCP Vegas 吞吐率比較-DCCP 先進入

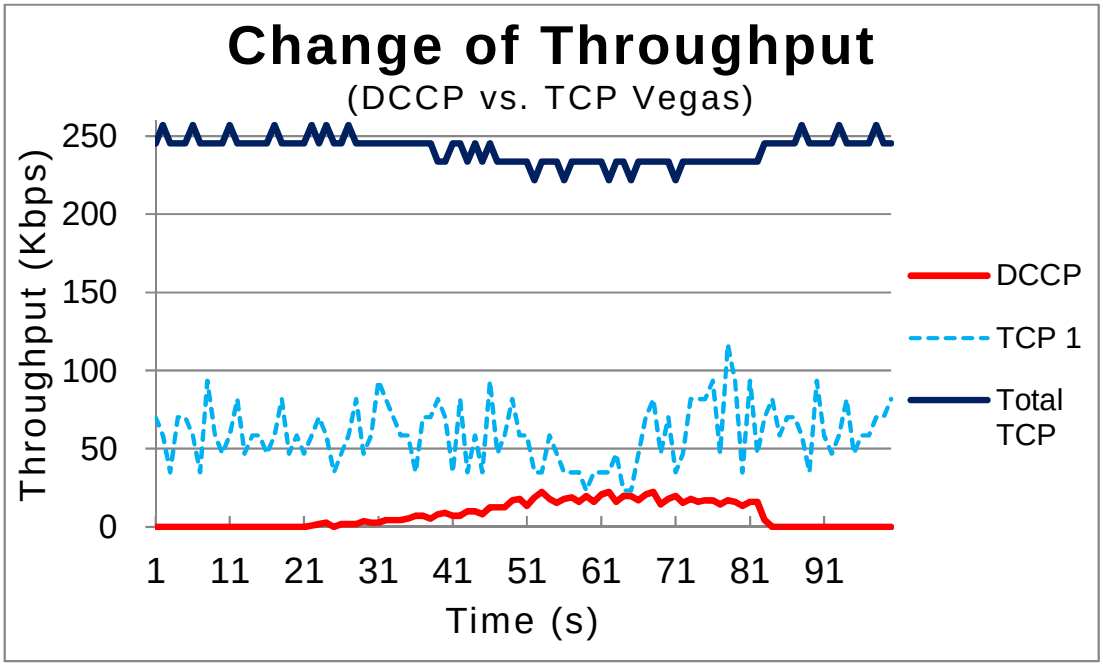

圖 23　DCCP 與 TCP Vegas 吞吐率比較-TCP Vegas 先進入

總之當 DCCP 面對 TCP Tahoe、Reno、NewReno、SACK 甚至是多條 Vegas 的競爭之下都處於劣勢。表 6 中列出以 DCCP 為基礎的 VoIP 串流在 10-20 秒、21-30 秒、31-40 秒和 41-50 秒之間的平均吞吐率，其中基準值(100%)是在 0-10 秒間網路沒有 TCP 資料流存在。

表 6　DCCP 吞吐率-DCCP 先進入

Period(sec)	10-20	20-30	30-40	40-50	50-60
vs. Tahoe	25.12%	25.23%	25.47%	21.61%	16.59%
vs. Reno	29.67%	32.01%	23.01%	16.94%	20.21%
vs. NewReno	29.79%	25.82%	20.68%	21.26%	15.89%
vs. SACK	21.73%	17.64%	17.76%	15.77%	16.12%
vs. Vegas	72.78%	52.45%	33.18%	19.63%	12.62%

3.3 VoIP 通話品質分析

本節中討論 VoIP 的通話品質，在圖 24,25,26,27,28,29,30,31 中顯示，以 UDP 為基礎的 VoIP 通話品質不受影響；而 DCCP 會因越來越多的 TCP 資料流進入網路，通話品質越來越糟。另外表 7 也列出 DCCP 的封包延遲和封包遺失率，其中封包遺失包含送達接收端但時間超過 VoIP 所能容忍的範圍。使用 UDP 的 VoIP 能維持 300 毫秒以下的封包延遲，且在 TCP 競爭下依然能維持 20%以下的封包遺失率；DCCP 相較之下就顯得遜色許多，面對 TCP 的競爭，封包延遲時間越來越長，封包遺失率越來越高，甚至超過了 80%，而且當只有一條 TCP 資料流時，吞吐率降至初始值的 30%。

DCCP 在面對 TCP Vegas 的競爭下，情況稍好一點，圖 32,33 中顯示，使用 DCCP 的 VoIP 在一條 Vegas 的競爭下勉強能維持通話品質，封包延遲時間保持在 200 毫秒下、封包遺失率維持在 27%以下。

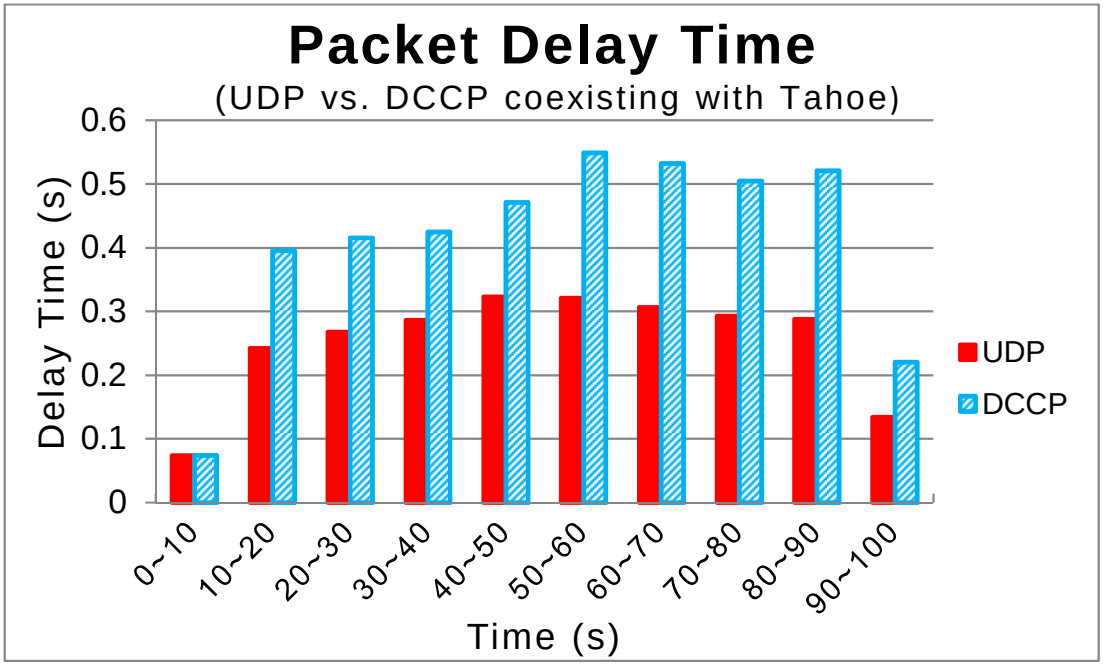

圖 24　UDP 與 DCCP 平均封包延遲比較-與 TCP Tahoe 競爭

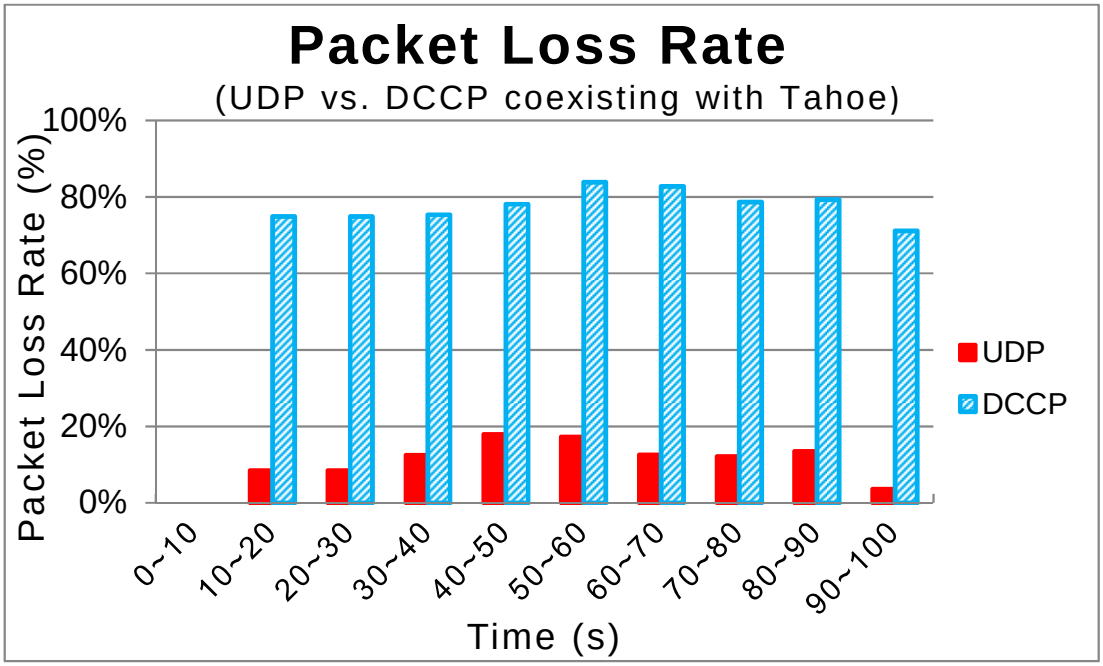

圖 25　UDP 與 DCCP 平均封包遺失率比較-與 TCP Tahoe 競爭

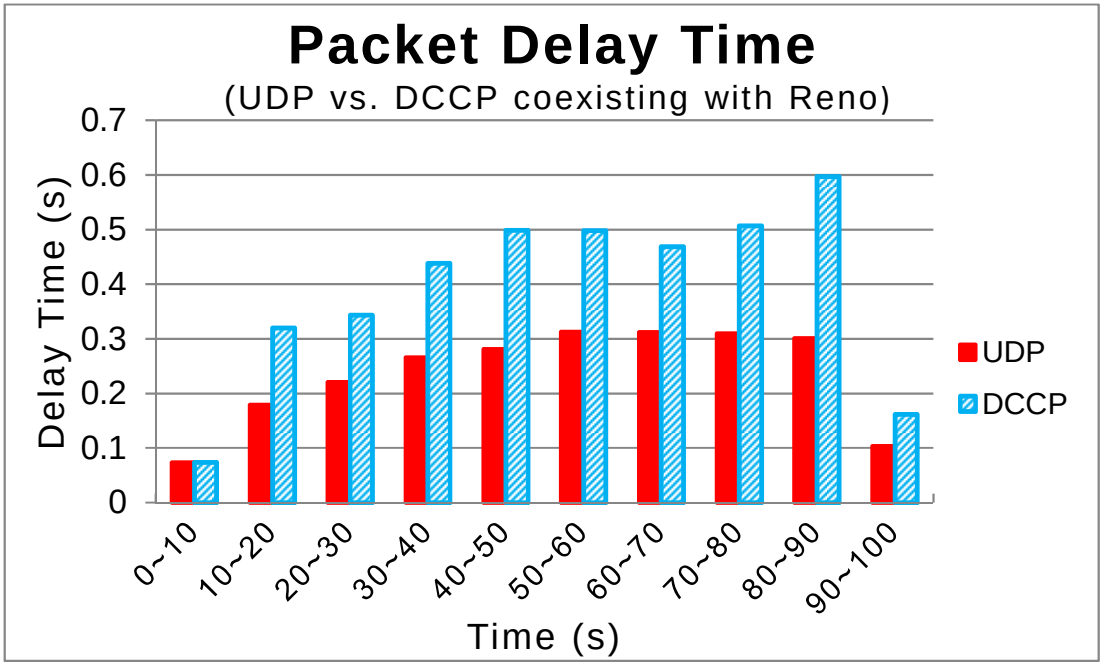

圖 26　UDP 與 DCCP 平均封包延遲比較-與 TCP Reno 競爭

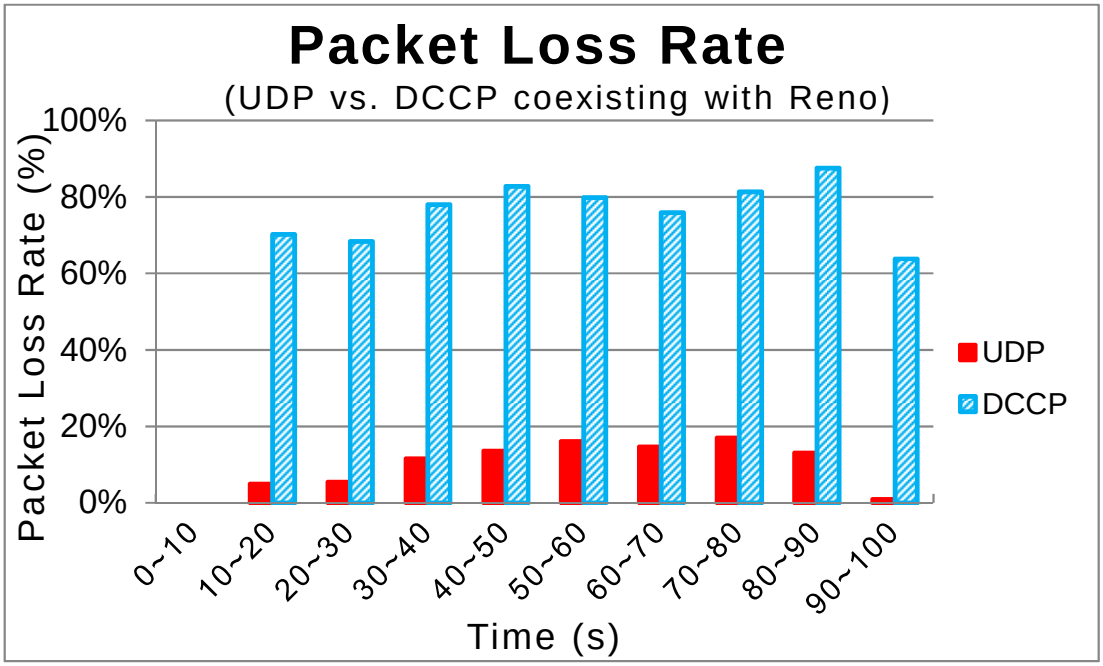

圖 27　UDP 與 DCCP 平均封包遺失率比較-與 TCP Reno 競爭

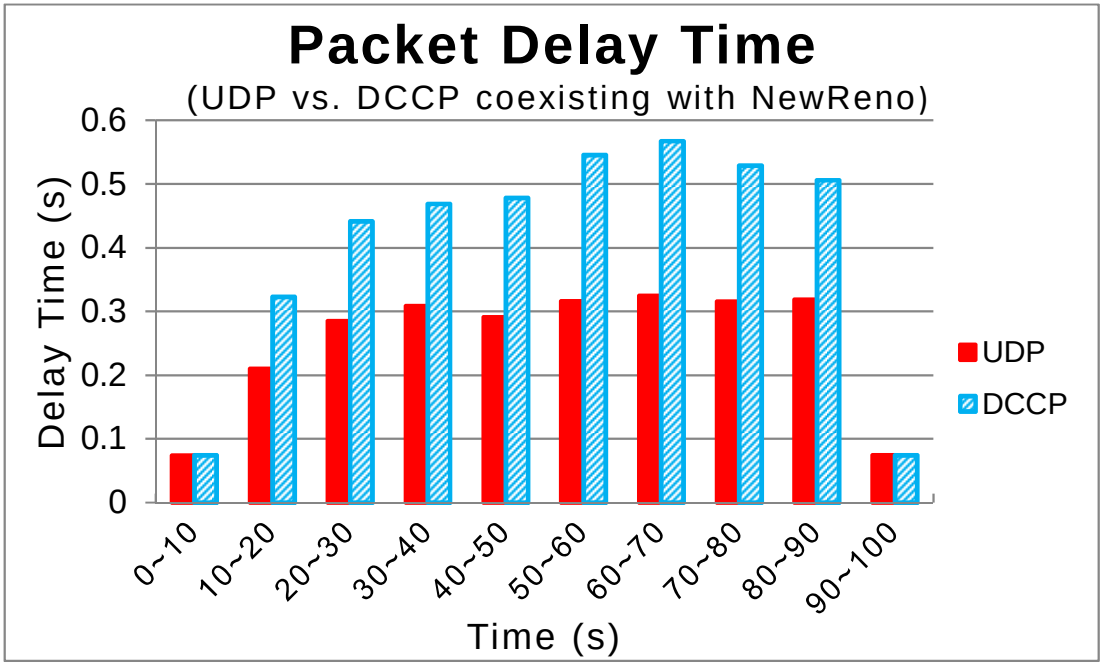

圖 28　UDP 與 DCCP 平均封包延遲比較-與 TCP NewReno 競爭

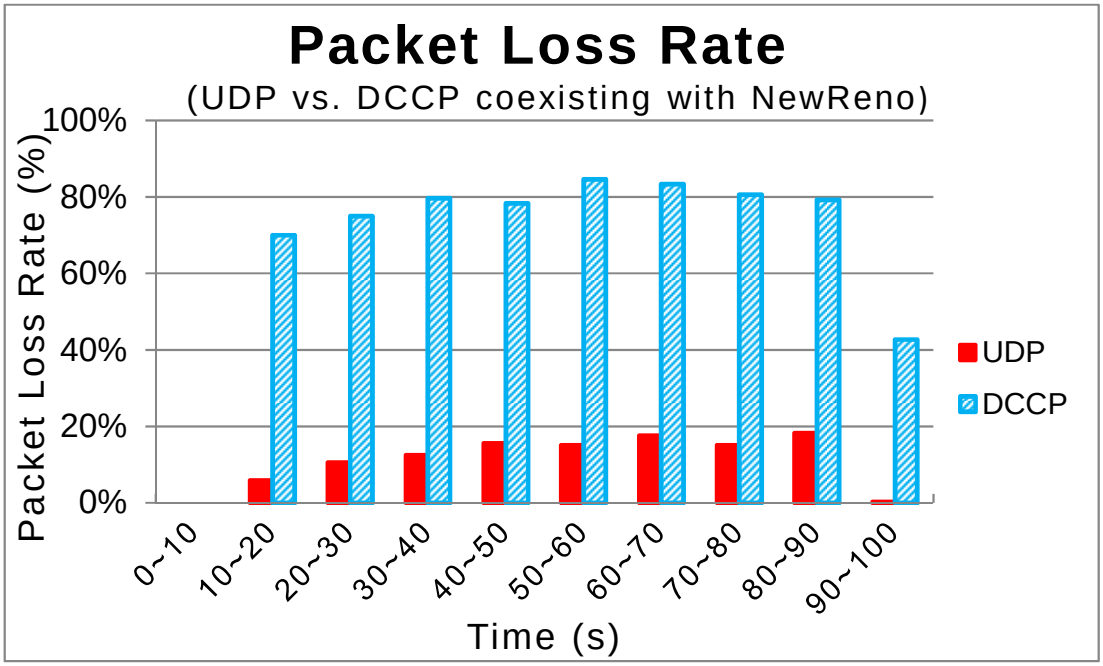

圖 29　UDP 與 DCCP 平均封包遺失率比較-與 TCP NewReno 競爭

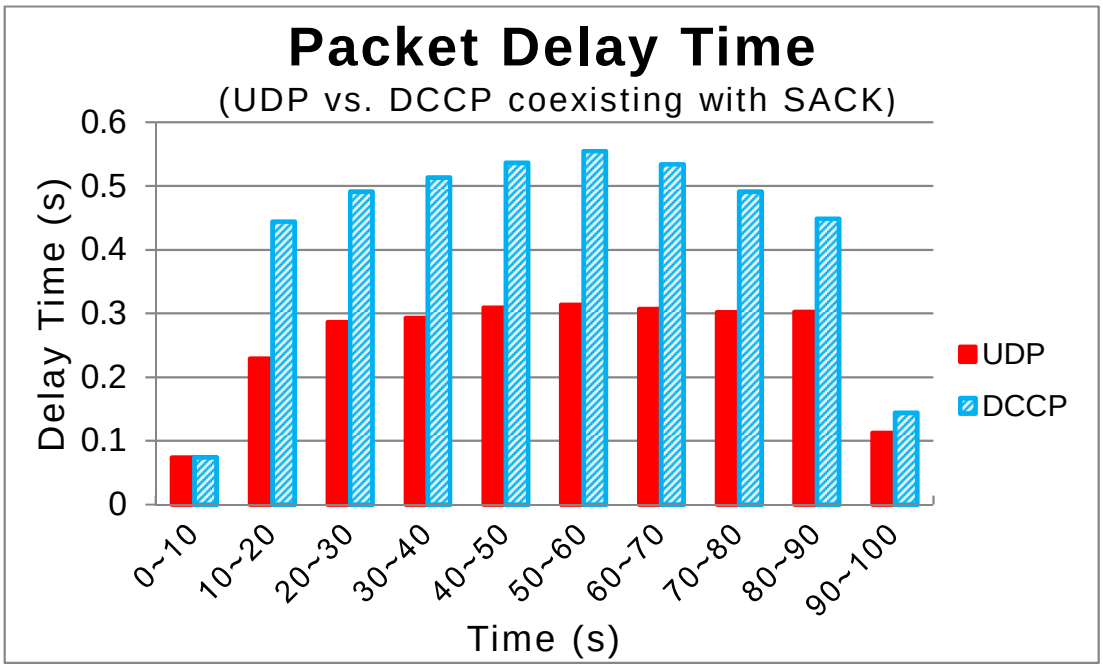

圖 30　UDP 與 DCCP 平均封包延遲比較-與 TCP SACK 競爭

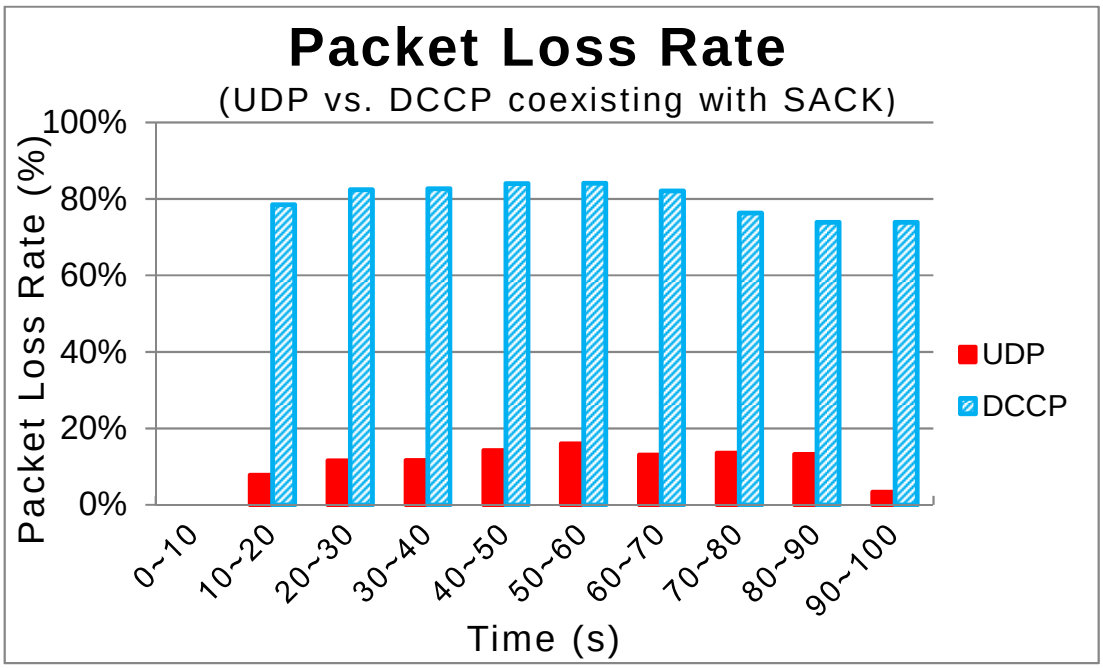

圖 31　UDP 與 DCCP 平均封包遺失率比較-與 TCP SACK 競爭

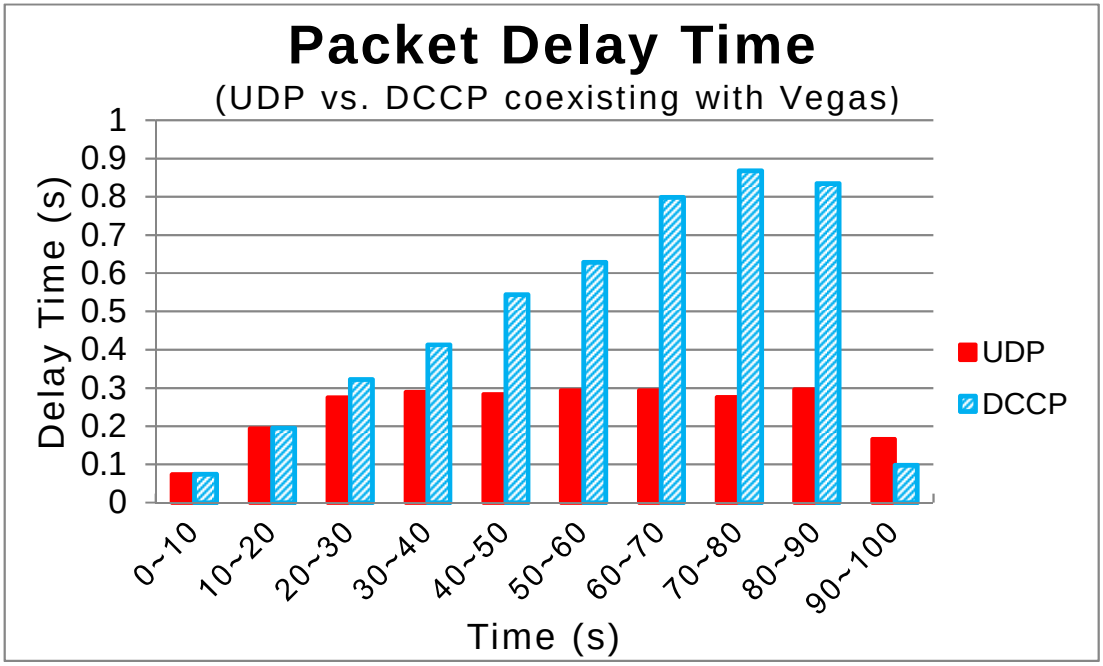

圖 32　UDP 與 DCCP 平均封包延遲比較-與 TCP Vegas 競爭

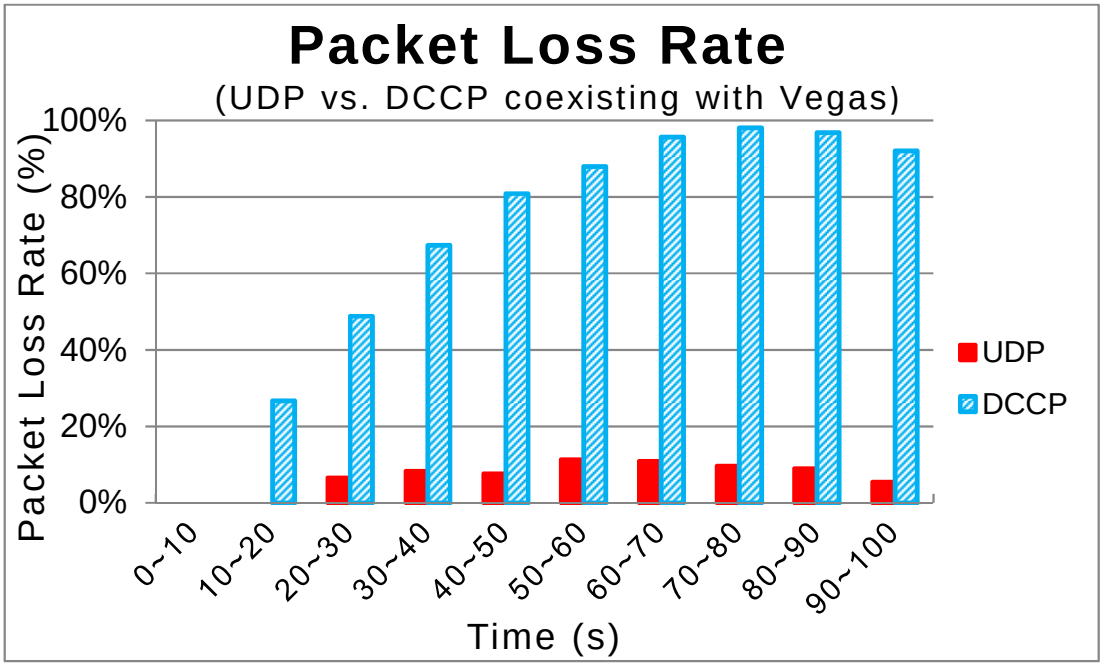

圖 33　UDP 與 DCCP 平均封包遺失率比較-與 TCP Vegas 競爭

表 7　DCCP 為基礎的 VoIP 通話品質

Period (sec)	0-10	10-20	20-30	30-40	40-50	50-60
vs. Tahoe						
Throughput Ratio(%)	100	25.12	25.23	25.47	21.61	16.59
Delay Time(ms)	73	395	415	424	470	548
Loss Rate(%)	0	74.86	74.86	75.32	78.09	83.85
vs. Reno						
Throughput Ratio(%)	100	29.67	32.01	23.01	16.94	20.21
Delay Time(ms)	73	319	343	438	498	497
Loss Rate(%)	0	70.13	68.40	77.97	82.81	79.82
vs. NewReno						
Throughput Ratio(%)	100	29.79	25.82	20.68	21.26	15.89
Delay Time(ms)	73	322	441	468	477	545
Loss Rate(%)	0	70.01	74.97	79.58	78.32	84.54
vs. SACK						
Throughput Ratio(%)	100	21.73	17.64	17.76	15.77	16.12
Delay Time(ms)	73	443	490	513	536	554
Loss Rate(%)	0	78.43	82.35	82.70	83.97	84.08
vs. Vegas						
Throughput Ratio(%)	100	72.78	52.45	33.18	19.63	12.62
Delay Time(ms)	73	194	321	412	543	627
Loss Rate(%)	0	26.64	48.79	67.36	80.85	87.89

3.4 小結

DCCP 是否能取代 UDP 是個大問題，我們利用 NS-2 模擬器模擬 DCCP 在各種 TCP 版本之下的表現。模擬結果顯示，DCCP 與目前常見的 TCP 版本在頻寬競爭的議題下尚有值得改進的地方，使用 DCCP 傳送封包的 VoIP 無法維持其傳輸效能，甚至於在 TCP 版本中侵略性最低 Vegas 之競爭下也無法有令人滿意的表現。雖然實驗結果無法代表真實網路中的所有情況，但在我們的實驗拓樸是一個典型的越洋網路電話環境，這是目前網路電話最常見的情境。

第四章 調整封包大小的壅塞控制方法

很多時效性應用必須以固定間隔時間送出封包，縮短封包在網路中的延遲時間。目前的 VoIP 在通話前會先設定好傳輸速率，固定其 bit-rate，在整個傳輸的過程中不再改變，可變速率方法(Flexible Bit-rate)能在通話的過程中，在網路頻寬足夠時使用較低的縮率提高語音品質，在網路頻寬不足的情況下使用較高的壓縮率來調整 bit-rate 以避免網路壅塞。

圖 34 中 original packet stream 是編碼器以固定時間間隔(例如 30 毫秒)送出一系列的語音封包，第二個 change inter-packet time 表示將編碼器產生的封包置於 buffer 中，在壅塞控制的指揮下送出資料流，因為 buffer 的緣故，inter-packet time 是變動的；第三個是採用固定封包間隔而變動封包大小的方法送出之資料流，圖中顯示其送出時間與編碼器產生封包的時間幾乎同步，因此可知 change inter-packet time 的方式會導致延遲時間拉長，但 change packet size 則否。

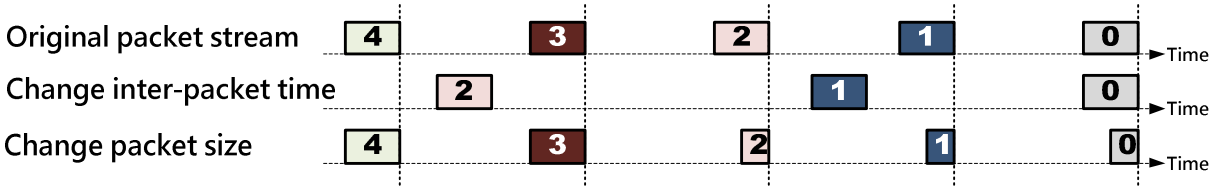

圖 34　調整發送間隔與調整封包大小之壅塞控制方式比較

4.1 設計目標

目前 DCCP 無法取代 UDP 成為不可靠傳輸協議的主流，且現行的壅塞控制機制不適用於即時性網路應用程式，但我們仍可以配合應用程式來改變語音封包的大小，藉此調整傳輸速率。

4.1.1 手動調整方法

我們提出一套網路電話架構，讓使用者能在網路電話通進行中，藉由自身對於通話品質的感受，手動的調整語音壓縮 bit-rate 來適應網路情況。

4.1.2 自動調整方法

我們提出一套網路電話架構，讓系統能在通話進行中偵測到網路壅塞的情況，當網路壅塞時自動降低語音壓縮 bit-rate 來舒緩頻寬，等到網路回復順暢時自動提高語音壓縮 bit-rate 提升通話品質。

4.2 Bit-rate 調整演算法

4.2.1 壅塞偵測方法

網路電話是使用定位元速率(Constants Bit-rate)方式傳送語音封包，封包傳送到接收端時，連續收到兩個封包的間隔時間應該在一個固定的時間範圍之內；當頻寬不足時，接收端連續兩個封包的間隔會拉長，偵測到此現象時視為網路壅塞，如圖 35 所示。

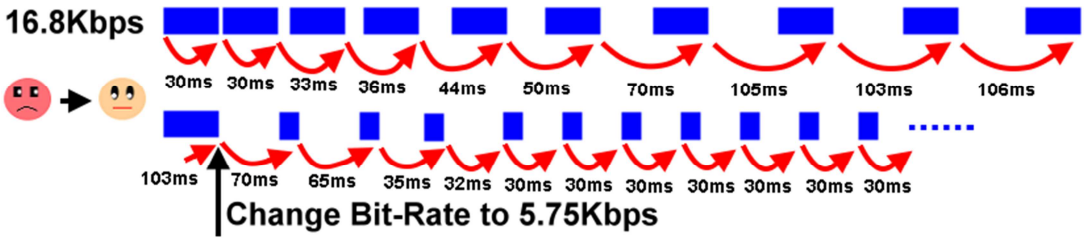

圖 35　改變 Bit-rate 以適應網路情況

我們假設連續兩個封包的間隔時間為t_i，且服從常態分配，則兩封包間隔時間可用$t_1, t_2, \ldots, t_N$表示，其平均值為$\bar{t}$(4)，標準差為σ_t(5)，並且為避免每次都重新計算標準差，我們可以將(5)推導成(6)。

$$\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{4}$$

$$\sigma_t = \sqrt{\frac{1}{N-1}\sum_{i=1}^{n}(t_i - \bar{t})^2} \tag{5}$$

$$\sum_{i=1}^{N}(t_i - \bar{t})^2 = \sum_{i=1}^{N}({t_i}^2 - 2t_i\bar{t} + \bar{t}^2) = \sum_{i=1}^{N}{t_i}^2 - 2\bar{t}\sum_{i=1}^{N}t_i + \sum_{i=1}^{N}\bar{t}^2$$

$$= \sum_{i=1}^{N}{t_i}^2 - 2N\bar{t}^2 + N\bar{t}^2 = \sum_{i=1}^{N}{t_i}^2 - N\bar{t}^2 \tag{6}$$

$$\sigma_t = \sqrt{\frac{1}{N-1}(\sum_{i=1}^{N}{t_i}^2 - N\bar{t}^2)}$$

接下來依照常態分配之經驗法則，我們可以估計出：

- 下一個封包間隔時間t_i在$[\bar{t} - \sigma_t, \bar{t} + \sigma_t]$範圍之機率為 68.3%。
- 下一個封包間隔時間t_i在$[\bar{t} - 2\sigma_t, \bar{t} + 2\sigma_t]$範圍之機率為 95.4%。
- 下一個封包間隔時間t_i在$[\bar{t} - 3\sigma_t, \bar{t} + 3\sigma_t]$範圍之機率為 99.7%。

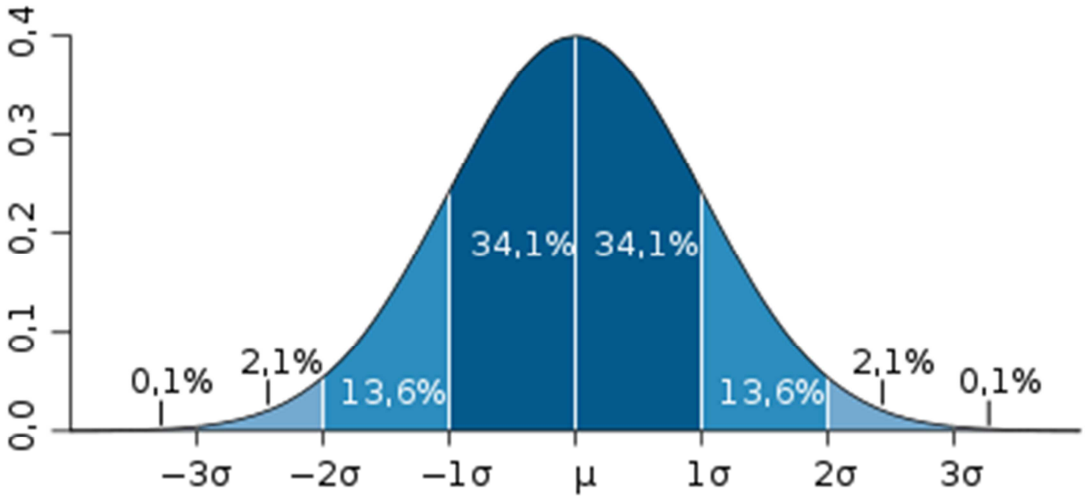

圖 36　常態分配之經驗法則

4.2.1 降低 Bit-rate 演算法

利用常態分配之經驗法則，並避免不斷的調整 bit-rate，我們設計下一個封包間隔時間t_{new}小於$[\bar{t}+3\sigma_t]$視為網路狀況正常，如果t_{new}超過$[\bar{t}+3\sigma_t]$則判斷為網路壅塞，連續超過α次則送出降低 bit-rate 一個等級的指令給發話端，演算法如下。

$when\ receive\ a\ new\ packet\ at\ T_i, Then\ t_{new} = T_i - T_{i-1}$
$if\ t_{new} \geq \bar{t} + 3\sigma_t\ and\ congestion\ frequency \geq \alpha$
　　$send\ decrease\ bitrate\ command\ to\ sender$
　　$congestion\ frequency = 0$
$else\ if\ t_{new} \geq \bar{t} + 3\sigma_t$
　　$congestion\ frequency = congestion\ frequency + 1$
$else$
　　$congestion\ frequency = 0$
　　$calculate\ new\ \bar{t}\ and\ new\ \sigma_t$
fi

4.2.2 提升 Bit-rate 演算法

當發話端收到受話端傳來的降低 bit-rate 指令後，此時發話端會進入 Low Bit-rate 狀態。為了避免發話端始終維持 Low Bit-rate 狀態，此時發話端每間隔β秒會提升發送語音 bit-rate 一個等級，直到 bit-rate 回到通話初始時的標準。

$$
\begin{aligned}
&when\ sender\ entry\ low\ bitrate\ state\\
&until\ [bitrate_{now} = bitrate_{start}]\\
&do\\
&\quad Sleep(\beta)\\
&\quad increase\ bitrate\\
&done
\end{aligned}
$$

4.2.3 調整 Bit-rate 範例

舉例而言，表 8 為十筆接收端收到語音封包的間隔時間，則在時間點T_7時可計算出平均間隔時間$\bar{t} = 33.6044$、標準差$\sigma_t = 1.646931$、$\bar{t} + 3\sigma_t = 38.54519$，我們看到在時間點$T_8$ 、T_9 、T_{10}時出現超過$\bar{t} + 3\sigma_t$的間隔時間，此時若α設定為 3，則受話端會向發話端送出降低 bit-rate 的命令。

表 8　以封包間隔時間調整 Bit-rate 範例

Time Spot	Inter-Packet Arrival Time
T_1	32.98
T_2	35.02
T_3	32.59
T_4	33.61
T_5	32.92
T_6	32.52
T_7	32.56
T_8	**39.41**
T_9	**38.58**
T_{10}	**38.58**

第五章 實驗與效能評估

本研究以實際網路上的實驗來評估可變速率網路電話與現行網路電話，藉由主觀指標 MOS 與客觀指標 E-Model，評估兩種方式在不同的網路環境之下的通話品質。

5.1 評估指標

- MOS(Mean Opinion Score)是直接利用人耳對於語音的感覺，並給予 1 到 5 分的分數。
- E-Model 使用網路的狀況，如 Packet Delay、Packet Loss、Jitter 和經編碼器(Codec)壓縮後的失真程度來評分，計算出對應到 1 至 4.5 分的 MOS。

5.2 實驗環境

在實驗一與實驗二中，我們利用兩台 PC 做為發話端和收話端，作業系統為 Windows 7，並安裝 Python 2.6 作為實驗程式環境；發送端與接收端之間透過 100MB 區域網路連結，並在另外一台 PC 上安裝 Ubuntu 10.04 當作路由器限制頻寬，作為網路瓶頸點。

實驗三的發話端 PC 使用中華電信光世代連上網際網路。光世代網路指在電信公司機房利用光纖迴路連接至光化交接箱，再利用乙太網路(Ethernet)或高速數位用戶迴路(VDSL)等技術，作為固網 IP 網路之接取電路，實驗中使用的下載速率約 10MBps、上傳速率為 2Mbps；另外收話端 PC 則透過台灣學術網路(TANet)連上網際網路。實驗中的網路瓶頸點在於中華電信光世代僅提供 2Mbps 的上傳速率，網路壅塞情形將在此發生。

由於 ITUT 所制定的標準編碼器(如 G.729 等)必須付費才使用，且只有固定的 bit-rate 可供選擇；故我們在實驗中所使用的編碼器為 Speex[23]，其最主要的特色是完全免費，且有 11 種不同大小的 bit-rate 可供選擇，符合我們實驗的要求，如表 9 所示。

表 9　Speex 編碼器各種 Bit-rate 相關參數

Quality	Bit-rate(Kbps)	Packet Size(Payload only)	MOS
0	3.95	10 Byte	2.258
1	5.75	15 Byte	2.7158
2	7.75	20 Byte	3.0106
3	9.8	25 Byte	3.3009
4	12.8	32 Byte	3.3009
5	16.8	42 Byte	3.5314
6	20.6	52 Byte	3.5314
7	23.8	60 Byte	3.6134
8	27.8	70 Byte	3.6134
9	34.2	86 Byte	3.9381
10	42.2	106 Byte	4.0925

5.3 實驗一

我們在實驗室的網路環境中探討在網路壅塞的情況下，透過提高語音壓縮率縮小語音封包，是否有助於語音通話品質，實驗拓樸如圖 37 所示。在發話端與接收端之間，透過路由器限制鏈結的頻寬為 5Kbps，封包延遲約在 1 毫秒以內。

圖 37　網路壅塞實驗拓樸

5.3.1 實驗結果與分析

我們在發話端分別使用不同的 bit-rate，傳送一段長度約 13 秒的語音至接收端，並記錄下封包傳輸的狀況於表 10。數據顯示在網路壅塞的情況下，使用較低 bit-rate 來傳輸 VoIP 的封包的遺失率較低；相反的使用較高 bit-rate 時則因為網路頻寬不足，導致封包遺失率增加。透過 MOS 分數可以得到一個結論，利用較低 bit-rate 來傳輸 VoIP，在網路壅塞的情況下確實能改善語音通話品質。

表 10　實驗一封包遺失率

Sender	Receiver	Bit-rate(Kbps)	Packet Loss Rate	MOS
326	326	7.75	0	3.01
326	326	9.8	0	3.30
326	305	12.8	0.0644	3.02
326	279	16.8	0.1285	2.15
326	254	20.6	0.2225	1.81

5.4 實驗二

當使用者覺得通話品質變糟時，可以手動調整 bit-rate。此實驗中我們從發話端傳輸一段語音至接收端，發話端傳輸時初始的語音壓縮 bit-rate 為 16.8Kbps，當接收端的使用者發現通話品質變糟後，手動調整語音壓縮 bit-rate 至 5.75Kbps。

5.4.1 實驗結果與分析

圖 38 中實線代表使用者對於通話品質的感受，虛線為 bit-rate 的變化情況。通話初始時因為網路壅塞的原因，bit-rate 為 16.8Kbps 的語音壓縮率通話品質相當差；然而當使用者發現通話品質不佳，手動將 bit-rate 調整為 5.75Kbps 後，卻有較佳的通話品質。因此得出一個結論，在網路壅塞時，透過改變 bit-rate，確實能有效提高通話品質。

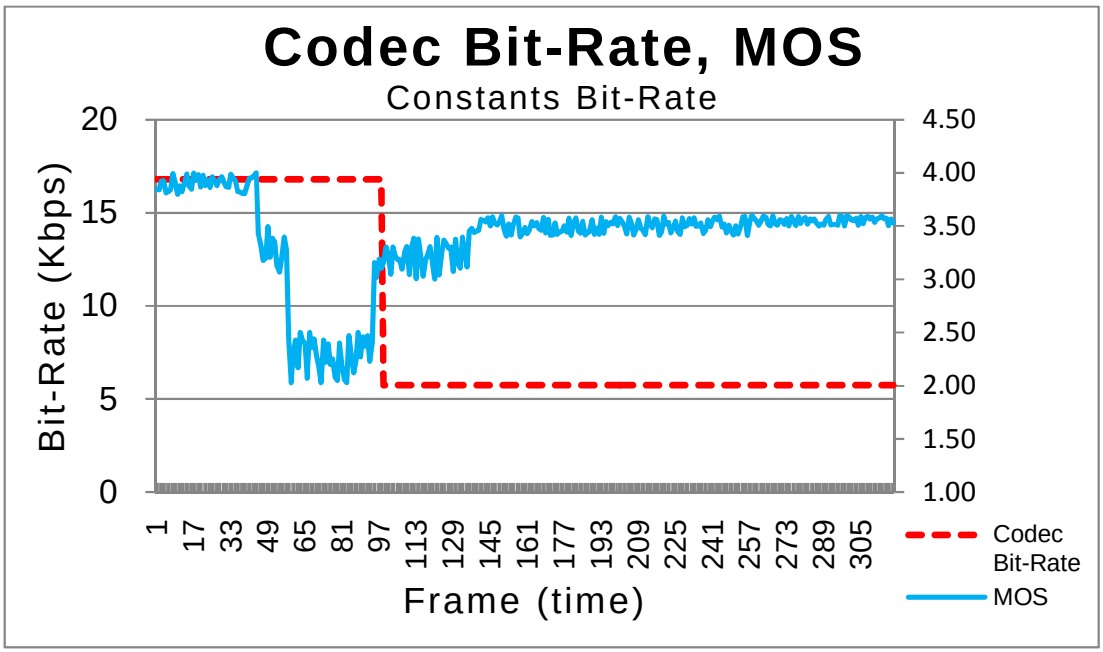

圖 38　手動調整 Bit-rate 之 MOS 變化

5.5 實驗三

在圖 39 的網路情境下，比較 UDP、DCCP 與可變速率方法傳送 VoIP 語音通話，觀察整體網路效能與通話品質，在實驗過程中，每隔 10 秒鐘增加 10 通電話進入網路中，直到網路中存在 100 條網路電話，並持續 20 秒後，之後每隔 10 秒鐘結束 10 通電話，等到 100 通電話都掛斷後結束實驗，實驗時間共 210 秒。

由於實驗的過程中送入高達 100 通 VoIP 語音通話，不可能用耳朵實際收聽所有電話，因此我們僅收聽一通電話並給予 MOS，其餘則利用在實驗過程中記錄下封包遺失率及 bit-rate 的變化，透過圖表呈現。

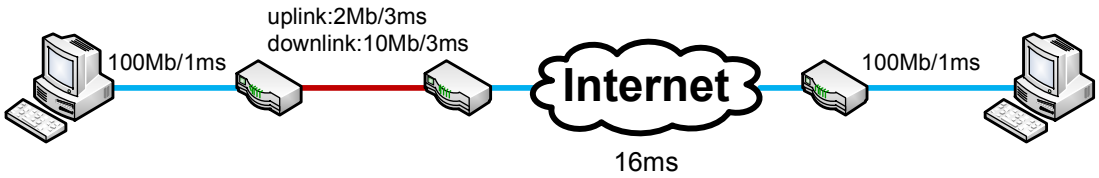

圖 39　實驗三網路拓樸

5.5.1 UDP-Based VoIP 實驗結果與分析

圖 40 中記錄著以 UDP 作為 VoIP 的傳輸協議時的封包遺失率與語音通話品質 MOS 的變化，圖 41 為傳輸 bit-rate 與 MOS 的變化。實驗中每隔 10 秒增加 10 通 VoIP 語音通話進入網路，直到網路中有著 100 通 VoIP 同時進行傳輸為止，實驗時間共 210 秒。

直到實驗時間 70 秒時，網路中共存在著 70 通 VoIP，此時封包遺失率近乎為零，這是因為網路尚能負載這些資料量。但隨著 VoIP 通話數增加至 80 通，封包遺失率逐漸增加，透過封包遺失率的變化我們可以發現，網路壅塞的情形越來越嚴重，僅僅透過 TCP 的壅塞控制機制已無法維持整體網路的穩定，語音通話品質也由原來的 MOS 4.1 分，下降至普通的 MOS 3.5 分。

當 VoIP 通話數到達 90 通時，網路壅塞加劇，封包遺失率增加至 15%，整體通話品質更降到糟糕的 MOS 2.4 分；當實驗時間 100 至 120 秒時，同時通話數到達 100 通時，整體封包遺失率已超過 20%，此時的 MOS 降到幾乎無法進行通話的 2 之下。

實驗時間 120 秒後，每隔 10 秒結束 10 通 VoIP 語音通話，直到所有語音通話結束後終止實驗。在這個過程中可以看到因為網路狀況逐漸舒緩，封包遺失率逐步下降，此時仍在通話中網路電話，其語音通話品質又慢慢回到網路頻寬充裕時的水準。

實驗數據詳細記載於表 11 與表 12，表中白色網底是封包遺失率介於 0%到 10%之間，在 VoIP 中可能還聽不太出來差異，淺色網底的是封包遺失率介於 10%到 20%之間，這是 VoIP 勉強還能容忍的範圍，深色網底的是封包遺失率大於 20%，已經嚴重影響到通話品質。

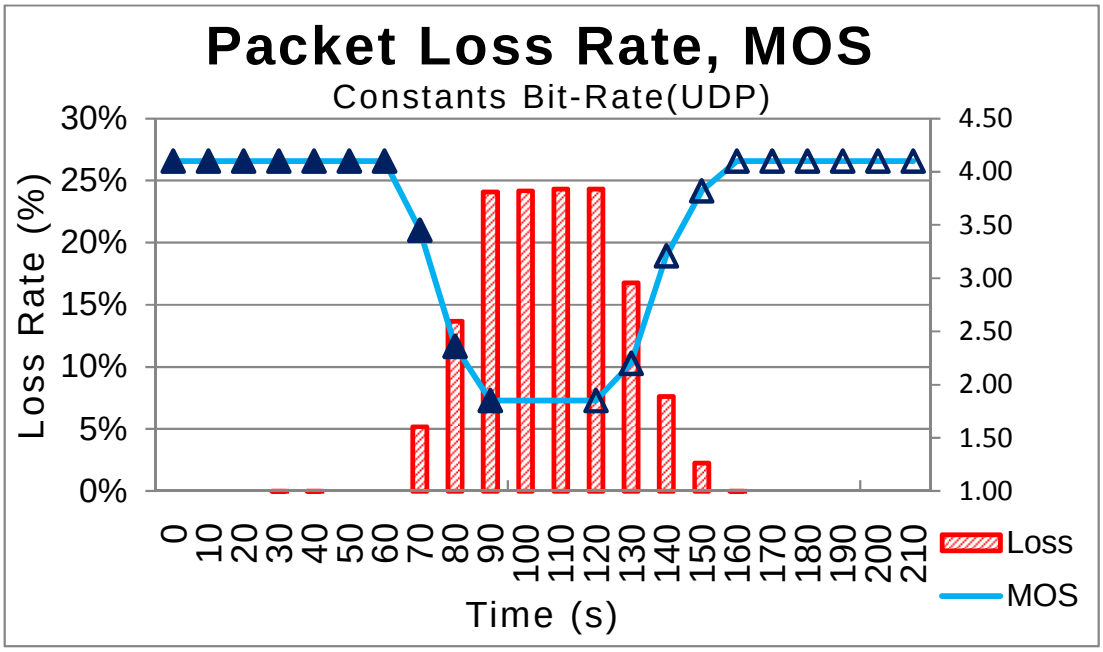

圖 40　Packet Loss Rate 與 MOS 變化-UDP

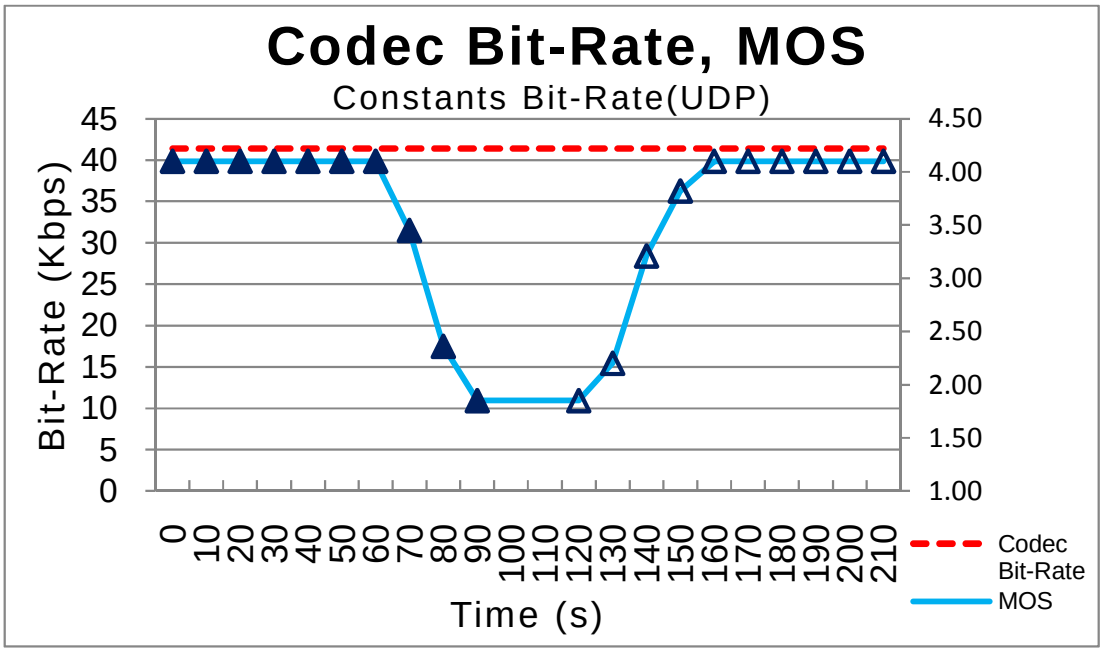

圖 41　Bit-rate 與 MOS 變化-UDP

表 11　以 UDP 傳輸 100 通 VoIP 封包遺失率之變化-PART1

Calls	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	Loss
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.06	0.05	0.06	0.03	0.05	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.04	0.04	0.04	0.04	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.37	0.11	0.09	0.12	0.06	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.46	0.22	0.21	0.20	0.12	0.08	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.07	0.07	0.04	0.05	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.45	0.39	0.48	0.36	0.21	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.07	0.07	0.07	0.02	0.06	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.01	0.34	0.40	0.52	0.30	0.22	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49	0.08	0.07	0.07	0.05	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.01	0.41	0.47	0.45	0.36	0.20	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11
11		0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.40	0.43	0.49	0.38	0.26	0.08	0.00	0.00	0.00	0.00	0.00	0.00		0.11
12		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.37	0.44	0.39	0.50	0.34	0.17	0.08	0.00	0.00	0.00	0.00	0.00	0.00		0.12
13		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.05	0.05	0.06	0.05	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.02
14		0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.23	0.29	0.21	0.26	0.21	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.07
15		0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.04	0.05	0.04	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.01
16		0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.35	0.38	0.47	0.47	0.35	0.17	0.08	0.00	0.00	0.00	0.00	0.00	0.00		0.12
17		0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.17	0.28	0.16	0.28	0.20	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.00		0.06
18		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.47	0.53	0.57	0.39	0.21	0.09	0.00	0.00	0.00	0.00	0.00	0.00		0.12
19		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.14	0.14	0.14	0.08	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.04
20		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.42	0.49	0.46	0.33	0.21	0.07	0.00	0.00	0.00	0.00	0.00	0.00		0.10
21			0.00	0.00	0.00	0.00	0.00	0.02	0.37	0.40	0.45	0.48	0.33	0.19	0.07	0.00	0.00	0.00	0.00	0.00			0.13
22			0.00	0.00	0.00	0.00	0.00	0.11	0.04	0.03	0.06	0.03	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.00			0.02
23			0.00	0.00	0.00	0.00	0.00	0.14	0.02	0.11	0.15	0.17	0.08	0.07	0.01	0.00	0.00	0.00	0.00	0.00			0.04
24			0.00	0.00	0.00	0.00	0.00	0.01	0.20	0.28	0.27	0.19	0.15	0.06	0.00	0.00	0.00	0.00	0.00	0.00			0.06
25			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.49	0.43	0.35	0.21	0.07	0.00	0.00	0.00	0.00	0.00			0.11
26			0.00	0.00	0.00	0.00	0.00	0.01	0.23	0.15	0.15	0.19	0.09	0.09	0.00	0.00	0.00	0.00	0.00	0.00			0.05
27			0.00	0.00	0.00	0.00	0.00	0.01	0.36	0.42	0.48	0.46	0.32	0.17	0.06	0.00	0.00	0.00	0.00	0.00			0.13
28			0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.07	0.07	0.07	0.06	0.01	0.00	0.00	0.00	0.00	0.00	0.00			0.02
29			0.00	0.00	0.00	0.00	0.00	0.02	0.38	0.47	0.42	0.46	0.34	0.23	0.04	0.00	0.00	0.00	0.00	0.00			0.13
30			0.00	0.00	0.00	0.00	0.00	0.01	0.05	0.06	0.04	0.05	0.07	0.02	0.00	0.00	0.00	0.00	0.00	0.00			0.02
31				0.00	0.00	0.00	0.00	0.14	0.01	0.49	0.62	0.47	0.43	0.25	0.11	0.00	0.00	0.00	0.00				0.16
32				0.00	0.00	0.00	0.00	0.02	0.19	0.13	0.13	0.16	0.12	0.02	0.00	0.00	0.00	0.00	0.00				0.05
33				0.00	0.00	0.00	0.00	0.01	0.01	0.09	0.12	0.12	0.08	0.01	0.00	0.00	0.00	0.00	0.00				0.03
34				0.00	0.00	0.00	0.00	0.02	0.01	0.19	0.19	0.19	0.12	0.08	0.00	0.00	0.00	0.00	0.00				0.05
35				0.00	0.00	0.00	0.00	0.00	0.46	0.06	0.08	0.04	0.05	0.03	0.00	0.00	0.00	0.00	0.00				0.04
36				0.00	0.00	0.00	0.00	0.05	0.48	0.18	0.21	0.17	0.12	0.07	0.00	0.00	0.00	0.00	0.00				0.08
37				0.00	0.00	0.00	0.00	0.12	0.01	0.05	0.07	0.02	0.04	0.01	0.00	0.00	0.00	0.00	0.00				0.02
38				0.00	0.00	0.00	0.00	0.05	0.00	0.40	0.45	0.45	0.33	0.21	0.06	0.00	0.00	0.00	0.00				0.12
39				0.00	0.00	0.00	0.00	0.15	0.00	0.08	0.11	0.10	0.04	0.04	0.00	0.00	0.00	0.00	0.00				0.03
40				0.00	0.00	0.00	0.00	0.01	0.49	0.19	0.19	0.19	0.13	0.09	0.01	0.00	0.00	0.00	0.00				0.08
41					0.00	0.00	0.00	0.13	0.22	0.27	0.24	0.24	0.46	0.29	0.14	0.00	0.00	0.00					0.14
42					0.00	0.00	0.00	0.02	0.02	0.03	0.06	0.05	0.04	0.01	0.00	0.00	0.00	0.00					0.02
43					0.00	0.00	0.00	0.00	0.00	0.07	0.07	0.04	0.06	0.02	0.01	0.00	0.00	0.00					0.02
44					0.00	0.00	0.00	0.01	0.00	0.10	0.09	0.08	0.06	0.02	0.00	0.00	0.00	0.00					0.03
45					0.00	0.00	0.00	0.05	0.18	0.27	0.19	0.23	0.24	0.07	0.00	0.00	0.00	0.00					0.09
46					0.00	0.00	0.00	0.14	0.01	0.11	0.12	0.08	0.04	0.03	0.00	0.00	0.00	0.00					0.04
47					0.00	0.00	0.00	0.14	0.02	0.20	0.13	0.12	0.09	0.08	0.00	0.00	0.00	0.00					0.06
48					0.00	0.00	0.00	0.14	0.01	0.11	0.07	0.11	0.04	0.03	0.00	0.00	0.00	0.00					0.04
49					0.00	0.00	0.00	0.14	0.14	0.28	0.14	0.22	0.18	0.04	0.02	0.00	0.00	0.00					0.08
50					0.00	0.00	0.00	0.00	0.38	0.11	0.08	0.14	0.04	0.02	0.00	0.00	0.00	0.00					0.06

表 12　以 UDP 傳輸 100 通 VoIP 封包遺失率之變化-PART2

Calls	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	Loss
51						0.00	0.00	0.03	0.01	0.41	0.46	0.53	0.34	0.20	0.08	0.00	0.00						0.17
52						0.00	0.00	0.00	0.00	0.11	0.08	0.12	0.05	0.03	0.00	0.00	0.00						0.03
53						0.00	0.00	0.17	0.16	0.23	0.32	0.26	0.17	0.09	0.02	0.00	0.00						0.12
54						0.00	0.00	0.01	0.11	0.05	0.04	0.05	0.04	0.02	0.00	0.00	0.00						0.03
55						0.00	0.00	0.15	0.24	0.30	0.22	0.29	0.47	0.27	0.13	0.00	0.00						0.17
56						0.00	0.00	0.01	0.07	0.08	0.02	0.06	0.04	0.03	0.00	0.00	0.00						0.03
57						0.00	0.00	0.17	0.50	0.17	0.21	0.18	0.12	0.06	0.00	0.00	0.00						0.12
58						0.00	0.00	0.00	0.48	0.06	0.03	0.06	0.04	0.02	0.00	0.00	0.00						0.06
59						0.00	0.00	0.01	0.02	0.39	0.46	0.50	0.35	0.21	0.06	0.00	0.00						0.17
60						0.00	0.00	0.14	0.06	0.07	0.06	0.05	0.03	0.01	0.00	0.00	0.00						0.03
61							0.00	0.01	0.38	0.48	0.46	0.48	0.34	0.17	0.06	0.26							0.26
62							0.00	0.12	0.00	0.52	0.46	0.56	0.42	0.28	0.12	0.25							0.27
63							0.00	0.01	0.00	0.14	0.21	0.20	0.13	0.06	0.01	0.19							0.09
64							0.00	0.02	0.24	0.23	0.19	0.25	0.14	0.10	0.00	0.17							0.14
65							0.00	0.16	0.00	0.11	0.10	0.10	0.06	0.03	0.00	0.16							0.07
66							0.00	0.02	0.20	0.28	0.21	0.22	0.16	0.04	0.02	0.16							0.13
67							0.00	0.02	0.38	0.45	0.47	0.43	0.31	0.16	0.06	0.13							0.24
68							0.00	0.02	0.24	0.31	0.24	0.24	0.13	0.07	0.01	0.12							0.14
69							0.00	0.18	0.00	0.12	0.08	0.11	0.03	0.01	0.00	0.08							0.06
70							0.00	0.01	0.18	0.28	0.24	0.24	0.10	0.08	0.00	0.06							0.12
71								0.11	0.00	0.06	0.04	0.06	0.06	0.01	0.49								0.10
72								0.12	0.19	0.20	0.29	0.34	0.16	0.06	0.53								0.24
73								0.02	0.37	0.10	0.12	0.09	0.04	0.03	0.47								0.15
74								0.00	0.00	0.41	0.44	0.45	0.30	0.21	0.50								0.29
75								0.00	0.02	0.04	0.05	0.05	0.02	0.03	0.42								0.08
76								0.01	0.02	0.44	0.51	0.43	0.34	0.19	0.46								0.30
77								0.00	0.01	0.05	0.06	0.06	0.04	0.01	0.36								0.07
78								0.17	0.01	0.44	0.50	0.47	0.34	0.20	0.41								0.32
79								0.00	0.02	0.04	0.03	0.05	0.02	0.01	0.32								0.06
80								0.00	0.03	0.15	0.15	0.17	0.11	0.07	0.29								0.12
81									0.17	0.22	0.21	0.14	0.15	0.79									0.28
82									0.00	0.45	0.46	0.48	0.35	0.88									0.44
83									0.25	0.34	0.18	0.17	0.17	0.75									0.31
84									0.00	0.05	0.09	0.10	0.07	0.71									0.17
85									0.01	0.05	0.07	0.05	0.04	0.69									0.15
86									0.23	0.25	0.28	0.13	0.14	0.69									0.29
87									0.16	0.22	0.21	0.14	0.15	0.79									0.28
88									0.00	0.45	0.46	0.48	0.35	0.88									0.44
89									0.25	0.34	0.18	0.17	0.17	0.75									0.31
90									0.00	0.06	0.09	0.10	0.07	0.71									0.17
91										0.45	0.49	0.47	0.98										0.60
92										0.45	0.43	0.52	0.96										0.59
93										0.49	0.47	0.47	0.96										0.60
94										0.08	0.05	0.05	0.91										0.27
95										0.50	0.50	0.51	0.96										0.62
96										0.27	0.27	0.23	0.96										0.43
97										0.44	0.50	0.52	0.91										0.59
98										0.50	0.58	0.51	0.96										0.64
99										0.50	0.58	0.51	0.91										0.62
100										0.68	0.65	0.46	0.96										0.69
Avg.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.14	0.24	0.24	0.24	0.24	0.17	0.08	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.07

5.5.2 DCCP-Based VoIP 實驗結果與分析

圖 42 中記錄著以 DCCP 作為 VoIP 的傳輸協議時的，封包遺失率與通話品質 MOS 的變化，圖 43 為傳輸 bit-rate 與 MOS 的變化。實驗中每隔 10 秒增加 10 通 VoIP 語音通話進入網路，直到網路中有著 100 通 VoIP 同時進行傳輸為止，實驗時間共 100 秒。

實驗初始時，10 通 VoIP 進入網路中，封包遺失率近乎為零，顯示網路頻寬相當充裕;當通話數來到 20 通時,封包遺失率約在 5%左右,尚能維持 MOS 3.5 分的通話品質。當同時通話數由 30 通上升到 70 通時,因為 DCCP 的頻寬競爭力較差而降低了傳輸速率，使得語音封包暫緩送出，封包遺失率約為 15%至 20%左右，MOS 也由尚可的 3 分降至不佳的 2.5 分。

隨著 VoIP 通話數增加至 80 通，網路壅塞的情形越來越嚴重，封包遺失率也越來越高，DCCP 估計出的傳輸速率更低，暫緩送出封包的延遲也更為加劇，傳送端至接收端的封包延遲時間已超過 VoIP 的上限，造成封包遺失率增加至 45%左右，此時的 MOS 降到無法進行通話的 2 分之下。

實驗時間 120 秒後，每隔 10 秒結束 10 通 VoIP 語音通話，直到所有語音通話結束後終止實驗。在這個過程中可以看到因為網路狀況逐漸舒緩，封包遺失率逐步下降。當網路中通話數剩下 30 通時我們發現，相較於實驗初始時網路中存在 30 通電話，其封包遺失率略高，這是因為 DCCP 估計出的傳輸速率成長較為平緩所致。

實驗詳細結果記錄於表 13 與表 14，表中白色網底是封包遺失率介於 0%到 10%之間,在 VoIP 中可能還聽不太出來差異,淺色網底的是封包遺失率介於 10%到 20%之間，這是 VoIP 勉強還能容忍的範圍，深色網底的是封包遺失率大於 20%，已經嚴重影響到通話品質。

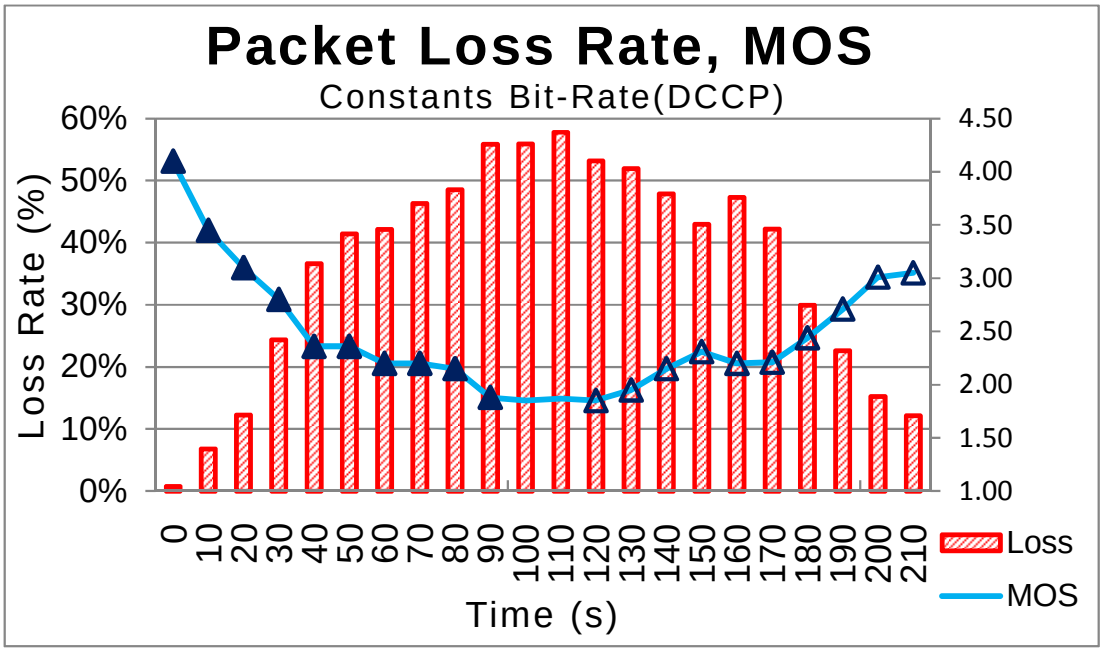

圖 42　Packet Loss Rate 與 MOS 變化-DCCP

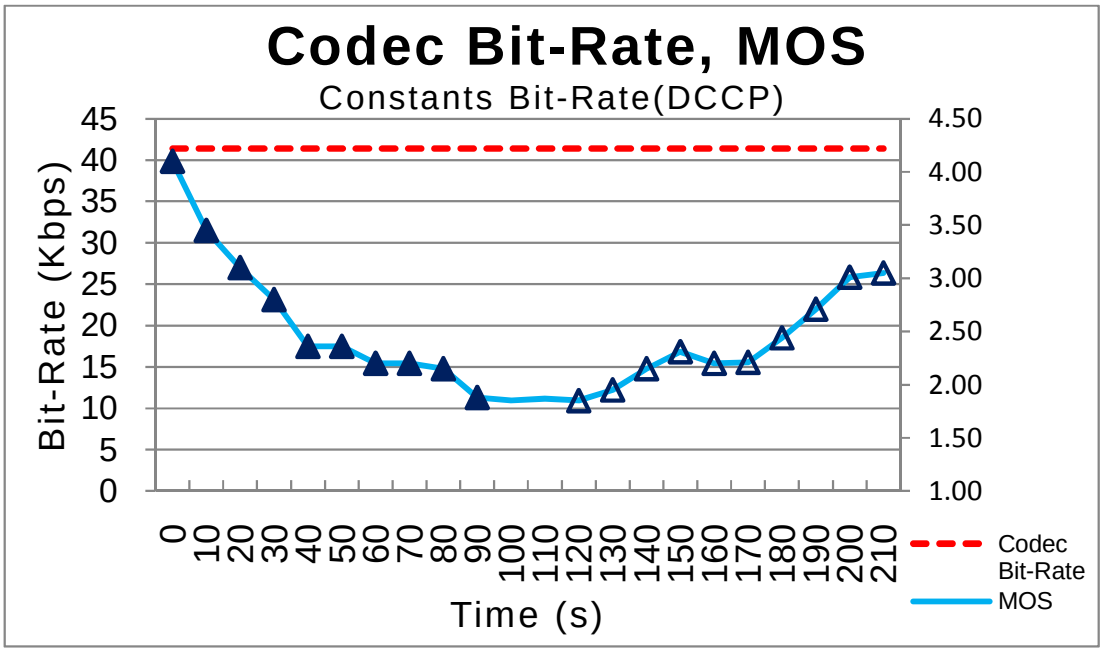

圖 43　Bit-rate 與 MOS 變化-DCCP

表 13　以 DCCP 傳輸 100 通 VoIP 封包遺失率之變化-PART1

Calls	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	Loss
1	0.00	0.06	0.10	0.39	0.51	0.34	0.42	0.47	0.47	0.59	0.58	0.62	0.57	0.55	0.50	0.43	0.40	0.48	0.31	0.22	0.12	0.08	0.40
2	0.01	0.08	0.15	0.18	0.25	0.50	0.35	0.45	0.47	0.60	0.59	0.55	0.50	0.50	0.46	0.41	0.52	0.36	0.35	0.27	0.14	0.12	0.38
3	0.01	0.05	0.13	0.15	0.51	0.43	0.42	0.41	0.46	0.58	0.62	0.57	0.51	0.55	0.50	0.40	0.41	0.36	0.27	0.21	0.20	0.10	0.38
4	0.01	0.08	0.10	0.30	0.49	0.33	0.43	0.53	0.49	0.54	0.59	0.57	0.58	0.51	0.44	0.45	0.47	0.42	0.31	0.27	0.20	0.09	0.39
5	0.01	0.06	0.14	0.43	0.26	0.43	0.39	0.40	0.46	0.55	0.50	0.58	0.57	0.52	0.49	0.46	0.51	0.45	0.27	0.26	0.15	0.11	0.39
6	0.01	0.07	0.10	0.07	0.29	0.50	0.49	0.42	0.50	0.52	0.59	0.60	0.52	0.51	0.45	0.44	0.50	0.39	0.27	0.25	0.17	0.14	0.37
7	0.00	0.09	0.10	0.25	0.29	0.49	0.48	0.43	0.44	0.57	0.55	0.60	0.50	0.52	0.45	0.45	0.53	0.40	0.34	0.28	0.21	0.12	0.39
8	0.01	0.10	0.13	0.23	0.43	0.46	0.45	0.51	0.51	0.57	0.63	0.60	0.52	0.50	0.48	0.45	0.43	0.40	0.32	0.18	0.15	0.17	0.40
9	0.01	0.07	0.14	0.21	0.53	0.50	0.49	0.41	0.48	0.58	0.56	0.59	0.49	0.54	0.51	0.45	0.47	0.47	0.27	0.19	0.14	0.10	0.40
10	0.00	0.11	0.11	0.25	0.33	0.34	0.41	0.47	0.47	0.54	0.62	0.63	0.50	0.51	0.46	0.41	0.42	0.36	0.29	0.22	0.16	0.18	0.37
11		0.05	0.12	0.10	0.24	0.38	0.35	0.52	0.47	0.59	0.59	0.54	0.50	0.53	0.48	0.41	0.41	0.41	0.27	0.28	0.08		0.38
12		0.09	0.14	0.09	0.34	0.44	0.39	0.53	0.51	0.54	0.58	0.54	0.57	0.51	0.48	0.42	0.48	0.40	0.31	0.18	0.12		0.40
13		0.05	0.12	0.32	0.36	0.34	0.44	0.43	0.52	0.60	0.52	0.55	0.51	0.54	0.50	0.42	0.52	0.47	0.30	0.23	0.22		0.41
14		0.07	0.11	0.25	0.42	0.41	0.42	0.51	0.50	0.52	0.60	0.54	0.54	0.53	0.50	0.44	0.42	0.48	0.27	0.26	0.07		0.41
15		0.05	0.12	0.08	0.34	0.34	0.44	0.43	0.44	0.57	0.48	0.53	0.58	0.50	0.49	0.43	0.45	0.40	0.27	0.21	0.22		0.38
16		0.09	0.14	0.19	0.32	0.50	0.38	0.48	0.46	0.60	0.47	0.61	0.52	0.54	0.46	0.40	0.46	0.42	0.33	0.26	0.12		0.40
17		0.07	0.13	0.30	0.52	0.37	0.47	0.40	0.45	0.56	0.57	0.59	0.56	0.49	0.50	0.43	0.47	0.42	0.33	0.22	0.17		0.41
18		0.03	0.10	0.34	0.23	0.45	0.50	0.49	0.45	0.52	0.60	0.59	0.52	0.49	0.45	0.44	0.50	0.48	0.28	0.21	0.07		0.40
19		0.09	0.13	0.24	0.24	0.35	0.35	0.53	0.46	0.51	0.52	0.58	0.50	0.51	0.49	0.39	0.43	0.38	0.28	0.24	0.18		0.38
20		0.00	0.13	0.23	0.35	0.33	0.37	0.42	0.48	0.58	0.47	0.56	0.50	0.52	0.50	0.46	0.45	0.39	0.30	0.20	0.12		0.38
21			0.15	0.32	0.29	0.41	0.47	0.42	0.50	0.54	0.52	0.56	0.56	0.55	0.49	0.45	0.49	0.39	0.28	0.22			0.42
22			0.11	0.15	0.26	0.48	0.33	0.47	0.44	0.55	0.53	0.61	0.54	0.55	0.46	0.41	0.44	0.46	0.27	0.20			0.40
23			0.11	0.29	0.28	0.37	0.35	0.50	0.52	0.52	0.55	0.63	0.49	0.50	0.47	0.41	0.41	0.44	0.27	0.17			0.40
24			0.15	0.33	0.36	0.35	0.37	0.41	0.49	0.61	0.62	0.58	0.50	0.54	0.44	0.43	0.46	0.42	0.28	0.25			0.42
25			0.14	0.15	0.36	0.46	0.45	0.46	0.45	0.52	0.62	0.63	0.56	0.51	0.47	0.42	0.43	0.43	0.30	0.25			0.42
26			0.10	0.12	0.28	0.33	0.43	0.50	0.48	0.54	0.63	0.63	0.49	0.52	0.50	0.45	0.52	0.36	0.35	0.24			0.41
27			0.12	0.43	0.41	0.35	0.34	0.52	0.50	0.60	0.53	0.59	0.54	0.50	0.50	0.41	0.46	0.41	0.29	0.27			0.43
28			0.12	0.43	0.31	0.35	0.50	0.39	0.52	0.55	0.48	0.61	0.50	0.53	0.44	0.46	0.48	0.44	0.31	0.17			0.42
29			0.11	0.33	0.25	0.50	0.48	0.42	0.44	0.56	0.50	0.54	0.49	0.50	0.50	0.43	0.43	0.48	0.29	0.20			0.41
30			0.12	0.43	0.30	0.36	0.34	0.48	0.51	0.51	0.51	0.59	0.51	0.54	0.45	0.41	0.51	0.42	0.33	0.18			0.42
31				0.27	0.52	0.36	0.46	0.47	0.48	0.59	0.48	0.55	0.50	0.50	0.50	0.46	0.51	0.49	0.29				0.46
32				0.32	0.31	0.33	0.45	0.44	0.53	0.56	0.51	0.63	0.56	0.56	0.45	0.42	0.46	0.39	0.32				0.45
33				0.13	0.40	0.49	0.45	0.52	0.53	0.56	0.56	0.62	0.55	0.50	0.47	0.44	0.51	0.49	0.32				0.47
34				0.21	0.53	0.40	0.39	0.45	0.53	0.51	0.63	0.55	0.57	0.53	0.44	0.46	0.50	0.41	0.34				0.47
35				0.08	0.47	0.45	0.39	0.41	0.48	0.54	0.62	0.59	0.56	0.51	0.50	0.40	0.43	0.42	0.33				0.45
36				0.31	0.31	0.48	0.48	0.50	0.45	0.58	0.51	0.60	0.57	0.52	0.50	0.40	0.46	0.46	0.31				0.47
37				0.09	0.40	0.46	0.43	0.51	0.50	0.53	0.61	0.53	0.53	0.55	0.44	0.41	0.46	0.46	0.29				0.45
38				0.24	0.40	0.48	0.46	0.50	0.52	0.52	0.64	0.55	0.49	0.49	0.47	0.41	0.42	0.42	0.31				0.46
39				0.40	0.41	0.42	0.44	0.47	0.50	0.51	0.48	0.55	0.57	0.54	0.45	0.44	0.53	0.48	0.29				0.47
40				0.10	0.24	0.44	0.43	0.49	0.44	0.57	0.56	0.55	0.50	0.53	0.50	0.39	0.53	0.46	0.29				0.44
41					0.45	0.48	0.33	0.43	0.53	0.57	0.56	0.55	0.52	0.50	0.50	0.46	0.51	0.36					0.48
42					0.42	0.35	0.34	0.52	0.46	0.53	0.48	0.62	0.54	0.53	0.50	0.43	0.49	0.45					0.48
43					0.42	0.45	0.41	0.52	0.51	0.51	0.58	0.55	0.56	0.50	0.45	0.46	0.51	0.46					0.49
44					0.46	0.35	0.34	0.46	0.48	0.61	0.49	0.57	0.53	0.54	0.46	0.44	0.53	0.42					0.48
45					0.33	0.43	0.36	0.42	0.53	0.59	0.62	0.59	0.51	0.51	0.46	0.40	0.53	0.43					0.48
46					0.44	0.46	0.48	0.46	0.46	0.57	0.63	0.58	0.55	0.49	0.50	0.46	0.44	0.40					0.49
47					0.38	0.37	0.46	0.53	0.52	0.61	0.53	0.54	0.57	0.55	0.45	0.46	0.47	0.41					0.49
48					0.26	0.49	0.40	0.45	0.51	0.52	0.62	0.59	0.56	0.53	0.51	0.43	0.41	0.37					0.47
49					0.30	0.35	0.37	0.47	0.44	0.57	0.51	0.55	0.51	0.51	0.48	0.44	0.47	0.36					0.45
50					0.51	0.38	0.36	0.50	0.50	0.56	0.58	0.55	0.50	0.53	0.51	0.41	0.49	0.37					0.48

表 14　以 DCCP 傳輸 100 通 VoIP 封包遺失率之變化-PART2

Calls	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	Loss
51						0.36	0.33	0.41	0.46	0.55	0.52	0.60	0.55	0.51	0.50	0.41	0.42						0.47
52						0.44	0.50	0.46	0.46	0.54	0.53	0.57	0.54	0.52	0.49	0.40	0.51						0.50
53						0.49	0.43	0.50	0.51	0.54	0.54	0.62	0.55	0.53	0.51	0.41	0.43						0.50
54						0.41	0.38	0.50	0.47	0.57	0.64	0.62	0.53	0.52	0.44	0.44	0.50						0.50
55						0.33	0.35	0.43	0.47	0.61	0.51	0.54	0.57	0.53	0.47	0.44	0.53						0.48
56						0.49	0.46	0.39	0.45	0.54	0.64	0.56	0.57	0.49	0.44	0.45	0.51						0.50
57						0.48	0.34	0.50	0.49	0.53	0.64	0.57	0.54	0.49	0.47	0.43	0.51						0.50
58						0.39	0.49	0.39	0.47	0.53	0.63	0.63	0.50	0.53	0.48	0.42	0.47						0.49
59						0.46	0.48	0.43	0.48	0.57	0.65	0.61	0.56	0.50	0.50	0.40	0.50						0.51
60						0.40	0.50	0.47	0.51	0.60	0.52	0.57	0.51	0.55	0.47	0.45	0.45						0.50
61							0.49	0.39	0.44	0.53	0.55	0.56	0.50	0.53	0.51	0.46							0.50
62							0.50	0.42	0.52	0.51	0.52	0.57	0.50	0.50	0.45	0.39							0.49
63							0.50	0.51	0.52	0.54	0.65	0.54	0.54	0.51	0.50	0.41							0.52
64							0.36	0.44	0.48	0.57	0.52	0.57	0.58	0.56	0.48	0.43							0.50
65							0.41	0.46	0.47	0.54	0.59	0.60	0.49	0.51	0.51	0.43							0.50
66							0.47	0.52	0.47	0.55	0.58	0.58	0.49	0.49	0.47	0.46							0.51
67							0.46	0.49	0.51	0.61	0.55	0.54	0.49	0.49	0.47	0.46							0.50
68							0.49	0.53	0.48	0.56	0.58	0.58	0.53	0.54	0.51	0.41							0.52
69							0.35	0.40	0.48	0.54	0.50	0.59	0.58	0.53	0.48	0.44							0.49
70							0.48	0.50	0.45	0.59	0.62	0.54	0.56	0.52	0.44	0.42							0.51
71								0.41	0.44	0.52	0.59	0.58	0.53	0.53	0.50								0.51
72								0.43	0.52	0.57	0.49	0.61	0.53	0.50	0.50								0.52
73								0.43	0.50	0.59	0.60	0.57	0.56	0.50	0.49								0.53
74								0.47	0.47	0.54	0.50	0.56	0.57	0.51	0.47								0.51
75								0.45	0.53	0.59	0.49	0.54	0.52	0.51	0.48								0.52
76								0.49	0.53	0.53	0.50	0.60	0.49	0.55	0.49								0.52
77								0.48	0.46	0.54	0.57	0.56	0.52	0.50	0.49								0.51
78								0.40	0.53	0.61	0.60	0.55	0.56	0.49	0.48								0.53
79								0.50	0.51	0.55	0.47	0.57	0.56	0.55	0.50								0.53
80								0.50	0.45	0.52	0.50	0.61	0.49	0.50	0.45								0.50
81									0.51	0.53	0.59	0.62	0.52	0.51									0.55
82									0.48	0.54	0.65	0.54	0.58	0.50									0.55
83									0.51	0.53	0.62	0.57	0.53	0.49									0.54
84									0.52	0.60	0.53	0.58	0.51	0.52									0.55
85									0.46	0.55	0.53	0.55	0.52	0.53									0.52
86									0.46	0.55	0.64	0.63	0.50	0.53									0.55
87									0.47	0.53	0.48	0.58	0.56	0.54									0.53
88									0.46	0.60	0.55	0.56	0.57	0.53									0.54
89									0.48	0.58	0.57	0.59	0.57	0.50									0.55
90									0.50	0.57	0.60	0.55	0.54	0.50									0.54
91										0.58	0.51	0.52	0.49										0.52
92										0.52	0.63	0.57	0.51										0.56
93										0.61	0.53	0.53	0.54										0.55
94										0.59	0.57	0.61	0.51										0.57
95										0.57	0.49	0.61	0.51										0.55
96										0.55	0.56	0.56	0.55										0.56
97										0.54	0.52	0.60	0.56										0.56
98										0.58	0.50	0.62	0.54										0.56
99										0.61	0.54	0.57	0.55										0.57
100										0.57	0.58	0.61	0.52										0.57
Avg.	0.01	0.07	0.12	0.24	0.37	0.41	0.42	0.46	0.48	0.56	0.56	0.58	0.53	0.52	0.48	0.43	0.47	0.42	0.30	0.23	0.15	0.12	0.38

5.5.3 Flexible Bit-rate VoIP 實驗結果與分析

圖 44 中記錄著我們所設計的 Flexible Bit-rate VoIP，封包遺失率與語音通話品質 MOS 的變化，圖 45 為傳輸 bit-rate 與 MOS 的變化。實驗中每隔 10 秒增加 10 通 VoIP 語音通話進入網路，直到網路中有著 100 通 VoIP 同時進行傳輸，實驗時間共 100 秒。

實驗初始時我們使用 42.2Kbps 的 bit-rate 來傳輸語音，當 10 通 VoIP 進入網路中，封包遺失率為零，通話品質也維持在良好的 MOS 4.1 分；隨著通話數成長到 30 通，網路偶爾發生壅塞，我們的演算法偵測到網路壅塞，降低了 bit-rate (23.8Kbps)來舒緩網路壅塞的情形，此時 MOS 保持在 3.5 分以上。

因通話數的成長，網路負載隨之加重，透過 Flexible Bit-rate 的方式來調節網路壅塞的情形，當同時通話數來到 80 通時，Flexible Bit-rate VoIP 仍可將封包遺失率控制在 2% 以下，維持一定的通話品質，直到網路中充滿了 100 通 VoIP 時，語音壓縮 bit-rate 調整至 16.8Kbps，通話品質仍能有 MOS 接近 3.5 分的水準。

實驗時間 120 秒後，每隔 10 秒結束 10 通 VoIP 語音通話，直到所有語音通話結束後終止實驗。在這個過程中可以看到因為網路狀況逐漸舒緩，平均 bit-rate 也隨之上升。當網路中通話數剩下 30 通時我們發現，相較於實驗初始時網路中存在 30 通電話，其封包遺失和語音壓縮 bit-rate (23.8Kbps)相差不遠，代表我們的壅塞控制機制在網路舒緩時，能快速反應，回復通話品質。

實驗詳細結果記錄於表 15 與表 16，表中白色網底是封包遺失率介於 0%到 10%之間，在 VoIP 中可能還聽不太出來差異，淺色網底的是封包遺失率介於 10%到 20%之間，這是 VoIP 勉強還能容忍的範圍，深色網底的是封包遺失率大於 20%，已經嚴重影響到通話品質。

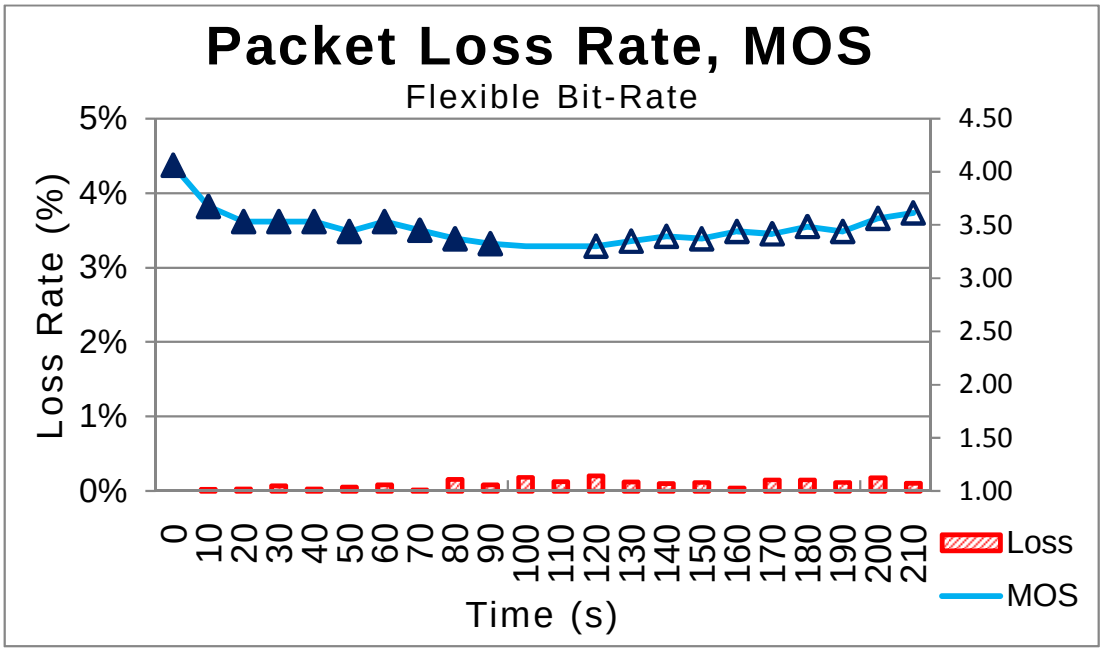

圖 44　Packet Loss Rate 與 MOS 變化-Flexible Bit-rate

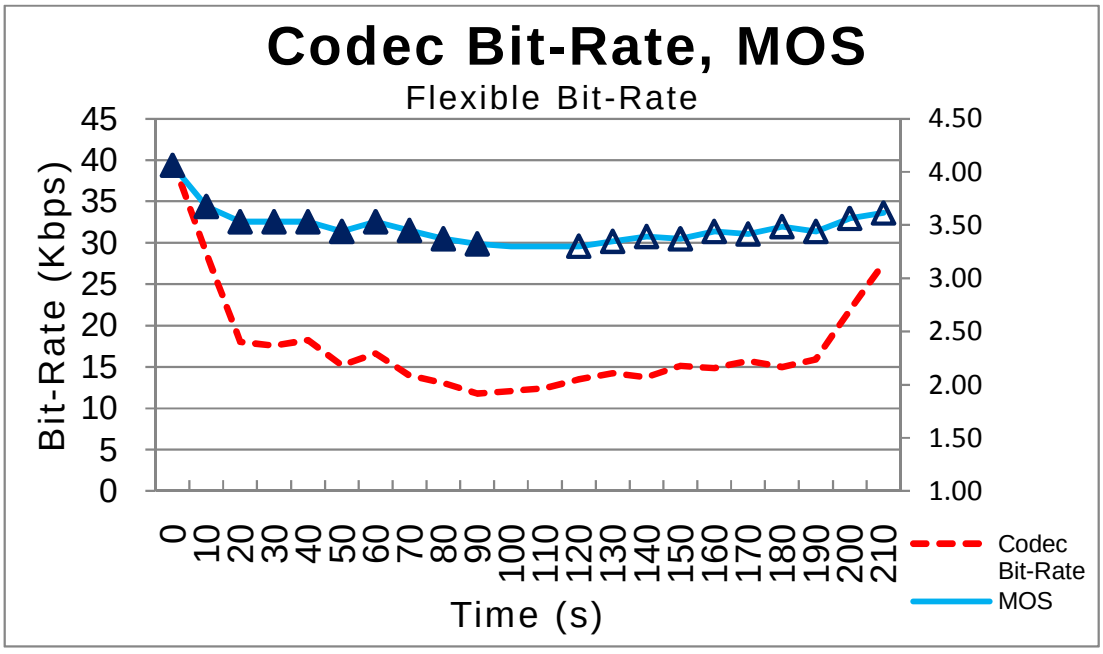

圖 45　Bit-rate 與 MOS 變化-Flexible Bit-rate

表 15　以 Flexible Bit-rate 傳輸 100 通 VoIP 封包遺失率之變化-PART1

Calls	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	Loss
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.02	0.00	0.00
10	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00		0.00
12		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00		0.00
13		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00
14		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00		0.00
15		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00		0.00
16		0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.01	0.00	0.00	0.00		0.00
17		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00
18		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00
19		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00		0.00
20		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01		0.00
21			0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.01			0.00
22			0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			0.00
23			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			0.00
24			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00			0.00
25			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			0.00
26			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00			0.00
27			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00			0.00
28			0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.02	0.00			0.00
29			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00			0.00
30			0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			0.00
31				0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00				0.00
32				0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.01				0.00
33				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00				0.00
34				0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00				0.00
35				0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.00				0.00
36				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00				0.00
37				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00				0.00
38				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00				0.00
39				0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00				0.00
40				0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00				0.00
41					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00					0.00
42					0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00					0.00
43					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00					0.00
44					0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00					0.00
45					0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.01					0.00
46					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00					0.00
47					0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01					0.00
48					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00					0.00
49					0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.00					0.00
50					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00					0.00

表 16　以 Flexible Bit-rate 傳輸 100 通 VoIP 封包遺失率之變化-PART2

Calls	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	Loss
51						0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00						0.00
52						0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.01	0.00	0.00	0.00						0.00
53						0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00						0.00
54						0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00						0.00
55						0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00						0.00
56						0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00						0.00
57						0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00						0.00
58						0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00						0.00
59						0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00						0.00
60						0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.01	0.00	0.00						0.00
61							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00							0.00
62							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00							0.00
63							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01							0.00
64							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00							0.00
65							0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00							0.00
66							0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00							0.00
67							0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00							0.00
68							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00							0.00
69							0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00							0.00
70							0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00							0.00
71								0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00								0.00
72								0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00								0.00
73								0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.00								0.01
74								0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00								0.00
75								0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00								0.00
76								0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00								0.00
77								0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01								0.00
78								0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00								0.00
79								0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00								0.00
80								0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00								0.00
81									0.01	0.01	0.01	0.00	0.00	0.00									0.00
82									0.00	0.00	0.00	0.00	0.01	0.03									0.01
83									0.00	0.00	0.00	0.00	0.00	0.00									0.00
84									0.00	0.01	0.01	0.00	0.00	0.00									0.00
85									0.00	0.00	0.00	0.00	0.00	0.01									0.00
86									0.00	0.00	0.00	0.00	0.00	0.01									0.00
87									0.00	0.00	0.00	0.00	0.00	0.00									0.00
88									0.00	0.00	0.00	0.00	0.00	0.00									0.00
89									0.00	0.00	0.00	0.00	0.02	0.02									0.01
90									0.00	0.00	0.00	0.01	0.00	0.00									0.00
91										0.00	0.00	0.01	0.01										0.00
92										0.00	0.00	0.00	0.00										0.00
93										0.00	0.00	0.00	0.00										0.00
94										0.00	0.00	0.00	0.00										0.00
95										0.00	0.00	0.00	0.00										0.00
96										0.00	0.00	0.00	0.00										0.00
97										0.00	0.00	0.00	0.00										0.00
98										0.00	0.00	0.00	0.00										0.00
99										0.00	0.00	0.00	0.00										0.00
100										0.00	0.00	0.00	0.00										0.00
Avg.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

5.5.4 三種傳輸方式實驗結果與分析

圖 46, 47 及表 17 顯示了 UDP、DCCP 以及 Flexible Bit-rate 三種方式傳輸 VoIP 的效能比較，分別為封包遺失率、通話品質(MOS)，以及頻寬效率。

結果顯示，在頻寬充裕的情況下，使用 UDP 傳輸封包的 VoIP，能有最佳的通話品質，但頻寬不足時將使得 VoIP 的封包遺失率增加，在實驗中的網路環境中能支持 70 通以 UDP 傳輸的 VoIP 進行高品質的語音通話，但當同時通話數量到達 100 通時，比起 10 通 VoIP 時，話品質指標 MOS 下降了 56.7%，封包遺失率增加了 25%。

而使用 DCCP 傳輸封包的 VoIP，其壅塞控制機無法與其他傳輸協議公平的分享頻寬，使得傳輸速率下降，語音封包的延遲加長而造成封包遺失。當通話數量到達 30 通時，比起只有 10 通 VoIP 時，MOS 下降了 24%，因延遲而造成的封包遺失率就已高達 13%；當網路中充滿 100 通 VoIP 時，比起初始時，封包遺失率增加了 44%。

使用 Flexible Bit-rate 來傳輸 VoIP 通話時，在網路中充滿 30 通 VoIP 通話時，偵測到輕微的網路壅塞而降低了 bit-rate，使得 MOS 下降了 13%，直到網路中充滿 100 通電話時，比起只有 10 通 VoIP 時，MOS 只下降了 17%，而整體的封包遺失率維持在 2%以下。經由實驗證明，比起使用 UDP 和 DCCP，以 Flexible Bit-rate 來傳輸 VoIP 通話時，可提高網路中 VoIP 的同時通話數，並維持整體網路封包遺失率低於 2%。

除了比較封包遺失率及 MOS 之外，我們另外定義了一個頻寬效率如(7)所示，表示單位頻寬所能獲得的 MOS。

$$Bandwidth\ Efficiency\ Ratio = \frac{MOS}{Goodput} \tag{7}$$

在表 17 中利用公式(7)計算出頻寬效率，在計算頻寬效率時，因 MOS 小於 2.5 幾乎無法進行通話，因此我們將 2.5 以下的頻寬效率視為 0；透過計算出的結果發現，以

Flexible Bit-rate 傳輸 VoIP 的頻寬效率高於以 UDP 和 DCCP 傳輸 VoIP，在網路壅塞的情況下尤其明顯。

另外透過 MOS 的平均值可以看到，以 UDP 和 Flexible Bit-rate 傳輸 VoIP 的 MOS 約為 3.5 左右，高於以 DCCP 傳輸 VoIP 的 2.46，這個結果表示以 DCCP 傳輸 VoIP 的通話品質較差。另一方面，以 UDP 傳輸 VoIP 的 MOS 標準差為 0.943，而以 Flexible Bit-rate 傳輸 VoIP 的 MOS 標準差為 0.168，這表示以 UDP 傳輸 VoIP 的 MOS 會隨著網路壅塞有較大幅的改變，而以 Flexible Bit-rate 傳輸 VoIP 的 MOS 較不會因為網路壅塞而有大幅的變動，能維持穩定的通話品質。

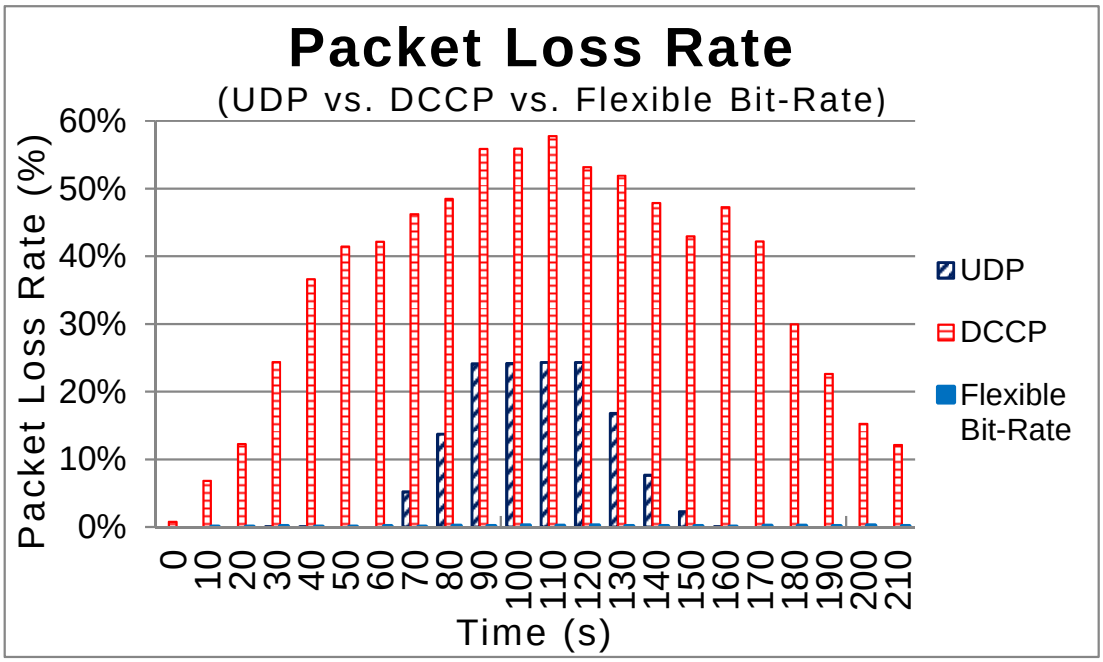

圖 46　三種方式傳輸 VoIP 的封包遺失率比較

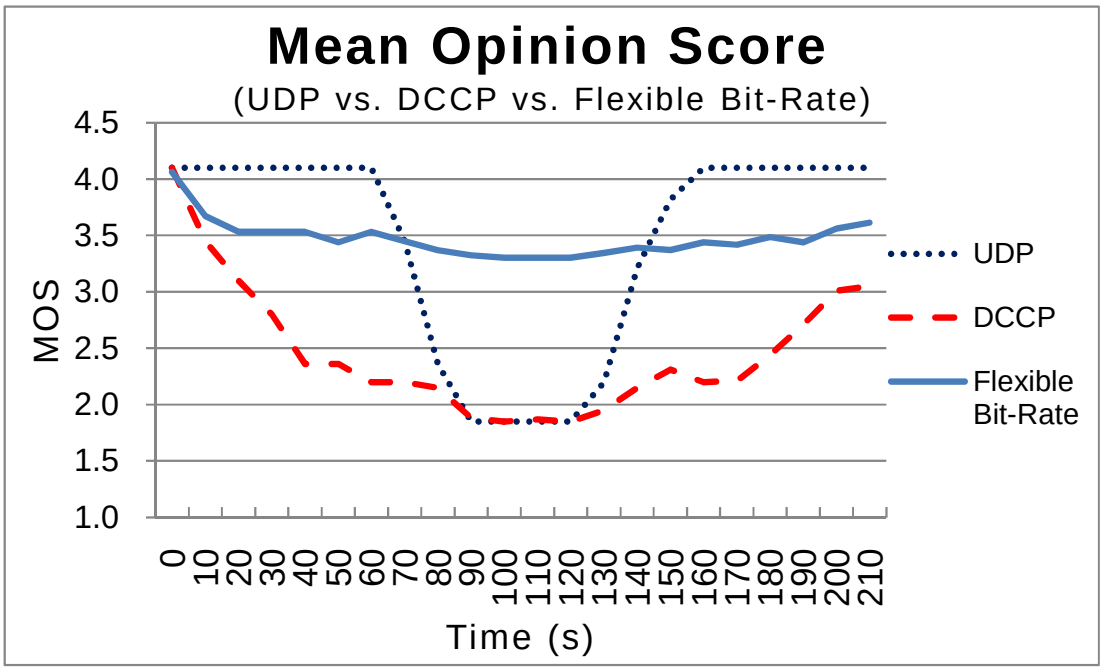

圖 47　三種方式傳輸 VoIP 的 MOS 比較

表 17　三種方式頻寬使用效率之比較

Calls	UDP			DCCP			Flexible Bit-rate		
	Goodput	MOS	Efficiency	Goodput	MOS	Efficiency	Goodput	MOS	Efficiency
10	44.35	4.10	0.092	43.64	4.10	0.094	42.52	4.06	0.096
20	44.33	4.10	0.092	38.48	3.45	0.090	34.38	3.67	0.107
30	44.33	4.10	0.092	34.12	3.10	0.091	21.80	3.53	0.162
40	44.32	4.10	0.093	25.35	2.80	0.110	20.99	3.53	0.168
50	44.31	4.10	0.093	17.79	2.36	0	20.41	3.53	0.173
60	44.34	4.10	0.092	15.20	2.36	0	17.62	3.44	0.195
70	44.31	4.10	0.093	14.84	2.20	0	18.89	3.53	0.187
80	39.83	3.45	0.087	12.80	2.20	0	16.22	3.45	0.213
90	32.99	2.36	0	11.75	2.15	0	16.10	3.37	0.209
100	25.52	1.85	0	8.63	1.88	0	15.01	3.32	0.221
100	25.48	1.85	0	8.62	1.85	0	14.34	3.30	0.230
100	25.36	1.85	0	7.90	1.87	0	14.57	3.30	0.227
100	25.38	1.85	0	9.73	1.85	0	17.18	3.30	0.192
90	30.71	2.20	0	10.25	1.95	0	16.34	3.35	0.205
80	37.85	3.21	0.085	12.04	2.15	0	15.90	3.39	0.213
70	42.34	3.82	0.090	14.42	2.31	0	17.63	3.37	0.191
60	44.32	4.10	0.093	12.32	2.20	0	16.64	3.44	0.207
50	44.33	4.10	0.092	14.81	2.21	0	18.62	3.42	0.183
40	44.34	4.10	0.092	21.76	2.44	0	17.74	3.49	0.196
30	44.30	4.10	0.093	26.53	2.71	0.102	18.43	3.44	0.187
20	44.30	4.10	0.093	31.85	3.01	0.094	25.74	3.56	0.138
10	44.33	4.10	0.092	34.23	3.05	0.089	31.18	3.61	0.116
Mean	39.17	3.44	0.07	19.41	2.46	0.03	20.38	3.47	0.18
S.D.		0.94			0.58			0.16	

5.6 小結

此章節裡我們探討了當網路頻寬不足時，各種 bit-rate 以及語音通話品質 MOS 的變化。從實驗一中發現，當網路頻寬不足時，利用較低 bit-rate 來傳輸 VoIP，遠比使用較高的bit-rate來傳輸VoIP能有較高的通話品質，起因於網路壅塞會造成高的封包遺失率，此時使用較高的 bit-rate 來傳輸 VoIP 將造成語音封包的遺失率增加，並對通話品質產生嚴重的影響；此外透過實驗二我們了解到，在通話的過程中遭遇網路頻寬不足的情形時，經由降低 bit-rate 的方式來傳輸 VoIP，能使得語音通話品質提升，有效的改善 VoIP 通話品質。

在實驗三中比較了 UDP、DCCP 及可變速率三種方式傳輸網路電話封包，觀察其封包遺失率及 MOS 變化，結果顯示利用可變速率方法，能有效降低網路電話的封包遺失率，維持通話品質。本實驗因係在實際網路中實測，除了實驗的網路電話之外，尚有其他資料流，其封包遺失率無法得知；但因網路對所有類型的封包一視同仁，網路電話的封包遺失率之升降，可以忠實的反應整體網路封效能的升降，故我們可以推測可變速率的壅塞控制機制，使得整體網路效能上升。

第六章 結論與未來研究

在 DCCP 這種具有壅塞控制機制的傳輸協議被提出後，我們期望它能取代 UDP 成為不可靠傳輸的主流協議，但透過 NS-2 網路模擬器和實際網路的實驗發現，DCCP 無法與其他傳輸協議公平的分享頻寬,也說明了現行DCCP的設計,尚無法完全取代UDP，且目前 DCCP 暫緩送出封包的壅塞控制機制,也不適用於講求時效性的網路應用程式。

本研究首先以實驗證明，當網路情況不佳時，DCCP 無法與其他傳輸協議公平的分享頻寬；當使用 DCCP 傳輸越洋長距離網路電話，如遇頻寬不足時，會因頻寬競爭力較弱而無法維持通話品質。本研究提出可變速率方法(Flexible Bit-rate)調整時效性網路服務的封包大小來進行壅塞控制，在維持一定服務品質之前提下，促進網路的和諧。我們在實際網路的實驗環境中評估以 UDP、DCCP 及可變速率三種方式傳輸網路電話封包的效能，結果顯示透過可變速率方法，能有效降低網路電話的封包遺失率，維持通話品質。

從研究結果得知，時效性的網路應用程式如果要提供壅塞控制機制的話，最好不要改變送出封包的間隔時間，而是改變其封包大小，過去調整 bit-rate 的觀點只針對 VoIP 自身的品質而言，並無考慮到整體網路的情況，且以往的 VoIP 數量不多，所以壅塞控制可能無關痛癢，但 VoIP 數量多時，就有責任必須參與壅塞控制。

本研究中僅粗略的對網路壅塞的情況進行壅塞控制，未來將針對不同的網路情況做更細微的壅塞控制。

參考文獻

[1]C. Albuquerque, B.J. Vickers, and T. Suda, "Network border patrol: Preventing congestion collapse and promoting fairness in the Internet," *IEEE-ACM Transactions on Networking*, vol. 12, no. 4, Dec. 2004, pp. 173-186.

[2]L. S. Brakmo, S. W. O'Malley, and Larry L. Peterson. "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *ACM SIGCOMM*, Aug. 1994, pp. 24-35.

[3]D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, vol.1, 1989, pp. 1-14.

[4]A. Falk, D. Katabi, Y. Pryadkin," Specification for the Explicit Control Protocol (XCP)," *draft-falk-xcp-03.txt (work in progress)*, July 2007.

[5]K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *ACM Computer Communication Review*, vol. 26, no.3, 1996, pp. 5-21.

[6]S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," *IETF RFC 2582*, 1999.

[7]S. Floyd and E. Kohler, "Profile for DCCP Congestion Control ID 2: TCP-like Congestion Control," *IETF draft-ietf-dccp-ccid2-08*, http://www.ietf.org/internet-drafts/draft-ietfdccp-ccid2-10.txt, Mar. 2005.

[8]S. Floyd, E. Kohler, and J. Padhye, "Profile for DCCP Congestion Control ID 3: TFRC Congestion Control," *IETF draft-ietf-dccp-ccid3-11*, http://www.ietf.org/internetdrafts/draft-ietf-dccp-ccid3-11.txt, Mar. 2005.

[9]Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," *IEEE INFOCOM*, San Francisco, CA, March 2003.

[10] Eddie Kohler, Mark Handley, and Sally Floyd, "Designing DCCP: Congestion Control Without Reliability," *SIGCOMM 06*, Sep. 2006, Pisa, Italy, pp. 27-38.

[11] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM*, Aug. 1988, pp. 314-329.

[12] N. E. Mattsson, "A DCCP module for ns-2," *http://epubl.luth.se/1402-1617/2004/175/LTU-EX-04175-SE.pdf*, Sep. 2004.

[13] J.Nagle, "Congestion Control in TCP/IP," *RFC896*, Jan 1984.

[14] K. Nahm, A. Helmy, and C.-C J. Kuo, "TCP over Multihop 802.11 Networks: Issues and Performance Enhancement," *ACM MobiHoc 05*, Urbana-Champaign, Illinois, USA, May 2005.

[15] F. Sabrina and J.-M. Valin,"Adaptive Rate Control for Aggregated VoIP Traffic," *GLOBECOM 2008*, pp. 1405-1410.

[16] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *IETF RFC 2001*, 1997.

[17] The E-Model, http://www.itu.int/rec/T-REC-G.107.

[18] "The Network Simulator - ns-2", http://www.isi.edu/nsnam/ns/.

[19] G.723, http://www.itu.int/rec/T-REC-G.723/e.

[20] G.729, http://www.itu.int/rec/T-REC-G.729/e.

[21] iLBC, http://www.ietf.org/rfc/rfc3951.txt.

[22] Mean Opinion Score, http://en.wikipedia.org/wiki/Mean_opinion_score.

[23] SpeeX, http://www.speex.org/.