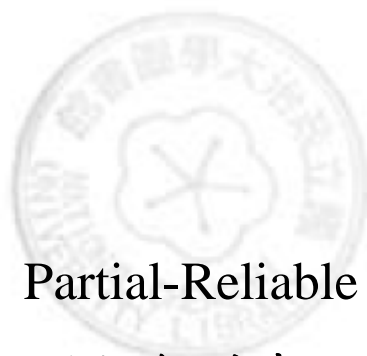


國立政治大學資訊科學系
Department of Computer Science
National Chengchi University

碩士論文

Master's Thesis



Partial-Reliable TCP

選擇性保證封包到達之通訊協定設計

研究生：吳明翰

指導教授：連耀南

中華民國九十七年三月

March 2008

選擇性保證封包到達之通訊協定設計

摘要

隨著網路的進步與發展，許多新興的資訊服務，如影音資訊，在網路上傳輸時並未要求封包都送達，不同的封包有不同的重要性，例如MPEG的關鍵畫面(I-Frame)就比其他畫面重要。常用的傳輸層通訊協定中，UDP與TCP都對所有的封包一視同仁，前者不做任何保證，而後者雖可保證所有封包的送達，但效率較差。

本研究提出一個新的TCP，"Partial-Reliable TCP"，使用選擇性重傳機制，配合應用程式的需求，對指定的封包提供遞送保護。當封包遺失時，只重傳保護的封包，可減少額外的網路資源消耗，並提升服務的品質。

此外，我們提出Single-Side的版本，接收端可以使用一般的TCP，在封包傳送時，讓接收端以為封包都是無誤傳達的，在server-client架構的網路服務中，只有伺服器端必須使用我們的Partial-Reliable TCP，大幅提高本通訊協定的可行性。

最後我們利用網路模擬工具NS-2來模擬實際網路環境，將我們的方法與現行的通訊協定在可解畫面封包數、PSNR值及額外耗用的網路資源三個參數做比較。我們使用兩個 Video 影像作為傳輸標的，在高遺失率的有線與無線網路的環境中進行實驗。當傳輸時間限制很短時，(例如影像會議的應用)，在有線的環境中，Basic PR-TCP比TCP Reno、TFRC最少增加約18%的可解畫面封包數，比UDP、TFRC及TCP Reno的PSNR值最少高出約15%，比TCP Reno及TFRC最少節省了12%的頻寬資源，Single-Side PR-TCP比Basic PR-TCP的PSNR值約低了11%，額外耗用的頻寬約多出10%。在無線的環境中，Basic PR-TCP比TCP Reno、TFRC最少增加約19%的可解畫面封包數，比UDP、TFRC及TCP Reno的PSNR值最少高出約20%，比Single-Side PR-TCP、TCP Reno及TFRC最少節省了15%的頻寬資源，Single-Side PR-TCP比Basic PR-TCP的PSNR值約低了14%。當傳輸時間限制較充裕時(例如VoD應用)，Basic PR-TCP雖然比TCP Reno及TFRC降低了約3%的PSNR值，但是最少能節省8%的頻寬耗費，Single-Side PR-TCP的PSNR值跟Basic PR-TCP相近，但是額外耗用約5%的頻寬資源。

Partial-Reliable TCP

Abstract

With the advance of computer and communication networks, many new information services over IP-based networks such as video streaming and VoIP (Voice over IP) are growing rapidly. These services can tolerate some packets lost in transmission without too much damage to their quality. The content carried in the packets of these services is not equally important in their replay processes. For example, key frames (e.g. I-Frames) of a video encoded in MPEG format are more important than others. The loss of I-frames may have a large impact to the quality of the transmitted video, while the loss of other types of frames may only have nominal damage. Unfortunately, the two most popular transport protocols, UDP and TCP, treat all packets equally without any discrimination. TCP guarantees the delivery of all packets, while UDP doesn't. TCP may waste too much resource to guarantee the delivery of unimportant packets, while UDP may fail to deliver too many important packets.

This thesis proposes a new TCP protocol, named Partial-Reliable TCP (PR-TCP), which applies selective retransmission strategy to

provide delivery guarantee to the selected packets designated by the application programs. In this way, we can save bandwidth consumption and reduce average delivery time without significant quality degradation. In fact, if the delivery of an object requires a stringent delivery time, the reduction of average delivery time may also lead to the reduction of abandoned packets at the receiver end. We propose two different versions of PR-TCP, Basic PR-TCP and Single-Side PR-TCP. Basic PR-TCP requires both ends of a connection to adopt PR-TCP while Single-Side PR-TCP only requires the sender end to adopt it. It is much easier to deploy Single-Side PR-TCP on the client-server systems where only servers need to use PR-TCP.

Finally, we use NS-2 network simulator to evaluate our PR-TCP against TCP Reno, TFRC and UDP. Two video stream samples are used for video sources. Three quality parameters are evaluated: wasted bandwidth consumption, PSNR, and the number of packets in decodable frames. Under heavy loaded wired network and short delay bound (<0.8 sec.), the simulation shows that Basic PR-TCP can outperform TCP Reno and TFRC in the number of packets in decodable frames by at least 18%. It can outperform TCP Reno, TFRC, and UDP in PSNR by at least 12%. The performance of Single-Side PR-TCP is less than Basic PR-TCP in terms of PSNR by 10%, and it consumes larger bandwidth by 8%. Under wireless environments where error rate is high, the simulation shows that Basic PR-TCP can outperform TCP Reno and TFRC in the number of packets in decodable frames by at least 19% as well as wasted transmission overhead by at least 15%. It can also outperform TCP Reno, TFRC, and UDP in PSNR by at least 20%. The performance of

Single-Side PR-TCP is less than Basic PR-TCP in terms of PSNR by 14%, and it consumes larger bandwidth by 10%. Under large delay bound (>8 sec.), the quality (PSNR) of the video transmitted using Basic PR-TCP is downgraded by only 3%, while it can save network bandwidth by 8%. The performance of Single-Side PR-TCP is about the same as Basic PR-TCP in terms of PSNR, but it consumes slightly larger bandwidth by 5%.



誌謝辭

本論文得以完成，要特別感謝連耀南教授兩年多來的指導，在專業知識、邏輯思考、表達能力及論文撰寫等，指導我許多，而且不厭其煩的一次又一次的指正我們，改正了我的做事態度及方法，真的很感謝。還要感謝蔡子傑教授及張宏慶教授平時的指導與協助。

另外，要感謝雷欽隆教授、陳耀宗教授、吳曉光教授及蔡子傑教授能在百忙之中撥冗參加我的論文口試，並指出可以改進加強的地方，使得本論文得以更加完善。

接著要感謝實驗室所有學長、同學及學弟，在這些日子能夠相互砥礪、相互共勉。最後要感謝我的家人朋友及所有幫助過我的人，謝謝！

吳明翰 March 18, 2008

目錄

第一章.....	1
1.1 UDP 簡介.....	3
1.2 TCP 簡介.....	4
1.3 多媒體影像在 IP 網路中傳輸之議題.....	4
1.4 研究動機與目的.....	5
1.5 論文組織架構.....	6
第二章.....	7
2.1 TCP 介紹.....	7
2.1.1 擁塞控制機制.....	7
2.1.1.1 TCP Tahoe and TCP Reno 的擁塞控制.....	8
2.1.1.2 慢啟動(Slow Start).....	9
2.1.1.3 擁塞避免(Congestion Avoidance).....	10
2.1.2 資料的封裝.....	13
2.2 Video streaming.....	14
2.2.1 Structure of video streaming.....	14
2.2.2 影像壓縮技術.....	15
2.2.3 MPEG (Motion Picture Experts Group)簡介.....	15
2.2.4MPEG 壓縮原理.....	17
2.3 相關研究.....	21
2.4 小結.....	24
第三章.....	25
3.1 問題分析.....	25
3.2 設計理念.....	25
3.3 Basic PR-TCP.....	26
3.3.1 Protection Class.....	26
3.3.2 Packet Life Control Scheme for Certified Packets.....	27
3.3.3 Selective Retransmission Scheme.....	27
3.3.4 Basic PR-TCP 的擁塞控制機制.....	28
3.3.5 小結.....	34
3.4 Single-Side PR-TCP.....	34
3.4.1 Protection Class.....	34
3.4.2 Replication Scheme.....	35

3.4.3 Single-Side PR-TCP 的擁塞控制機制.....	38
3.4.4 小結.....	42
3.5 Packet Protection by Forward Error Correction.....	42
3.6 小結.....	43
第四章.....	44
4.1 實驗環境.....	44
4.2 實驗評估指標 :.....	46
4.3 實驗設計.....	48
4.4 實驗 A:有線網路.....	49
4.4.1 實驗目標.....	49
4.4.2 實驗流程.....	50
4.4.3 實驗結果分析.....	51
4.5 實驗 B:有線網路+無線網路.....	57
4.5.1 實驗目標.....	57
4.5.2 實驗流程.....	58
4.5.3 實驗結果分析.....	59
4.6 實驗 C:影像品質評估.....	64
4.6.1 實驗目標.....	64
4.6.2 實驗流程.....	64
4.6.3 實驗結果分析.....	65
第五章.....	69
參考文獻.....	71

圖目錄

圖 1.1 Encoding Order and Display Order of Frames.....	5
圖 2.1 TCP Reno	9
圖 2.2 Packet Flow in Slow Start Phase.....	10
圖 2.3 Change of TCP Congestion Window Size.....	11
圖 2.4 User Data Forwarded Thru Layers of Protocols	13
圖 2.5 Video streaming over IP network	14
圖 2.6 Block of Frames in MPEG	16
圖 2.7 Encoding Percedure of MPEG	17
圖 2.8 Motion Estimation of MPEG	19
圖 2.9 Order of Edcode and Display per GOP.....	20
圖 3.1 Change of Basic PR-TCP Congestion Window Size	29
圖 3.2 State Diagram of Basic PR-TCP	30
圖 3.3 ACK Arrival Procedure in Baisc PR-TCP.....	32
圖 3.4 Pseudo Code for ACK Arrival Procedure in Basic PR-TCP.....	33
圖 3.5 No. of Replication V.S. Timeout Probability.....	35
圖 3.6 Single-Side PR-TCP Replication Scheme.....	37
圖 3.7 State Diagram of Single-Side PR-TCP	38
圖 3.8 ACK Arrival Procedure in Single-Side PR-TCP.....	40
圖 3.9 Pseudo code for ACK Arrival Procedure in Single-Side PR-TCP.....	41
圖 3.10 Packet Protection by Forward Error Correction.....	42
圖 4.1 Structure of Evalvid	45
圖 4.2 Simulation Video:Foreman 、Container.....	49
圖 4.3 Topology of Simulation A	50
圖 4.4 實驗 A 結果: 各通訊協定在不同 hop 數中 decodable packets 的比較.....	53
圖 4.5 實驗 A 結果: 各通訊協定在不同 hop 數中 PSNR 值的比較.....	54
圖 4.6 實驗 A 結果: 各通訊協定在不同 hop 數中額外耗用的網路資源比較.....	56
圖 4.7 實驗 A 結果: PR-TCP 比 TCP Reno 在 Saved resouce 及 PSNR 提升比較..	57
圖 4.8 Topology of Simulation B	58
圖 4.9 實驗 B 結果: 各通訊協定在不同 loss rate 中 decodable packets 的比較....	60
圖 4.10 實驗 B 結果: 各通訊協定在不同 loss rate 中 PSNR 值的比較.....	62
圖 4.11 實驗 B 結果: 各通訊協定在不同 loss rate 中 bandwidth wasted rate 的比較	63
圖 4.12 實驗 B 結果: PR-TCP 比 TCP Reo 在 Saved resource 及 PSNR 提升比較	64
圖 4.13 Original frame #111	65
圖 4.14 實驗 C 結果: Received Frame #111 by using TCP Reno	66
圖 4.15 實驗 C 結果: Received Frame #111 by using UDP.....	66

圖 4.16 實驗 C 結果: Received Frame #111 by using Single-Side PR-TCP67
圖 4.17 實驗 C 結果: Received Frame #111 by using TFRC.....67
圖 4.18 實驗 C 結果: Received Frame #111 by using Basic PR-TCP68



表目錄

表 1.1 Services classes	1
表 1.2 各服務類別之 QoS 品質要求.....	2
表 2.1 TCP 實作相關的 RFC 文件	8
表 3.1 Packet Loss Handling Procedure in Basic PR-TCP.....	28
表 3.2 State Transition in Basic PR-TCP	30
表 3.3 Replication overhead V.S. Timeout overhead	36
表 3.4 Packet Loss Handling Procedure in Single-Side PR-TCP.....	37
表 3.5 State Transition in Single-Side PR-TCP	39
表 4.1 PSNR and MOS	47
表 4.2 Parameters of Simulation A and B	49
表 4.3 Parameters of Simulation A.....	50
表 4.4 Parameters of Simulation B.....	58



第一章

簡介

隨著網路技術的進步與發展，更多新型的網路服務迅速應運而生，網路上所傳遞的不再只是資料及檔案，多媒體的應用服務像是動畫，影音在網路上的傳輸也越來越普及。這類型的資訊在網路上傳輸時並未要求每一個封包確實送達目的地，且不同的封包重要性也不同，例如 MPEG[34]壓縮格式的影音封包裡的關鍵畫面(I-frame)就比其他封包重要。常用的傳輸層通訊協定中，UDP[16]與 TCP[15]都對所有封包一視同仁，前者不做任何保證，而後者雖可保證所有封包的送達，但效率較差。

以下介紹目前比較廣為應用的資訊服務類型，UMTS[31]系統將網路上風行的主要應用依時效與品質需求概略分為四大類，如表 1.1，分別是交談式(Conversational class)，串流式(Streaming class)，互動式(Interactive class)，背景式(Background class)。

表 1.1 Services classes

Class	Example
Conversational	VoIP
Streaming	VoD
Interactive	Telnet
Background	E-Mail

表 1.2 各服務類別之 QoS 品質要求

	Delay Sensitivity	Jitter Sensitivity	Packet Loss Sensitivity
Conversational	High	High	Low
Streaming	Medium	High	Low
Interactive	Medium	Low	High
Background	Low	No	High

表 1.2[12]歸納出不同類型的服務對於 delay time、jitter 及封包遺失的需求，Conversational class 主要用來支援人類雙向溝通，根據人類感官之經驗歸納，此種服務對 long delay time 與 jitter 相當敏感，使用者在 delay time 超過 300 ms 時，就難以忍受其通話品質。Streaming class 則要求持續穩定的資料流，因此對 long delay time 與 jitter 也是相當敏感。Interactive class 與 Background class 都屬於 data communication 的服務，可容忍較長的資料傳送時間，但要求正確的資料傳送，因此幾乎無法忍受資料的遺失。

傳輸層通訊協定在傳輸路徑的兩個端點執行，負責將應用層交付的資料送達目的地，傳輸層協定必須負責將所交付的資料包裝成封包，交付給網路層，而在目的地重組為原來的資料。傳輸層協定必須負責決定最適當的傳送速率，必要時，並須保證資料完整無缺的送達目的地。最風行者有二，一為UDP、一為TCP，UDP協定非常簡單，它並不保證資料之完整，它負責將應用程式所交付的資料裝成封包後依指定的速度交付給網路層，由網路層送達目的地，而目的地的UDP則將所收到的封包直接交付給上層的應用程式，對於封包之遺失則置之不理，也不會依網路擁塞狀況調整傳送速率。

由於不須重送遺失的封包之故，UDP的整體傳送速率較快，因此對於時效性要求較高且不計較封包遺失的多媒體通訊最常使用。其弊病則因其不會依網路狀

況而調整傳送速率，而在網路擁塞時，不但遺失太多封包，且對網路有火上加油之弊，不免影響所傳送資料之品質。

另一方面，如果要保證資料之完整送達，最常使用的傳輸協定則非TCP莫屬，大部分的TCP協定依賴接收端送回的ACK[2]封包作為傳送封包的驅動信號，並據以預測網路的擁塞狀況以調整傳送速度(亦即調整Congestion Window大小)。當然要達到高可靠度的目的相對的就要付出一些代價，當封包遺失時，TCP便會啟動重傳機制，將遺失的封包重新傳送，當然重傳所付出的代價就是封包的延遲送達。這個機制在正常的網路下運作有不錯的效能，但是在某些特例情況下，則不能順利運作，最有名之例則是在無線網路下，封包較有線網路容易遺失時，TCP則會因誤判網路擁塞而過度降速，導致傳輸效能大幅降低。

Background class的服務，每個從傳送端送出的封包都必須無誤地送達接收端，所以其需要的是一個高可靠度的通訊協定，以確保資料都能正確無誤的送達，通常都是使用TCP來作為其通訊協定。

Streaming class的服務，時間的因素格外重要，封包延遲或遺失，會造成聲音或是畫面的破碎，降低服務品質。目前最常用來傳送即時性訊務流的通訊協定為UDP，雖然UDP不會因為重傳而導致封包延遲，但是UDP無法根據網路狀況決定傳送速度，當網路擁塞情況嚴重，封包遺失的數量過大，對即時性訊務流的服務品質會造成極大的影響。例如網路電話可容忍某種程度之封包遺失，但過度遺失則不免讓通話品質下降，又如在傳送video時如果遺失太多關鍵畫面(I-Frame)，則大幅影響畫面品質。

1.1 UDP 簡介

UDP (User Datagram Protocol)[16]是一種用於 IP 網路的通訊協定，其特色為非連接導向(Connectionless-Oriented)，資料在傳輸前不會先建立連線，且不會檢

查傳送出去的資料是否到達接收端，所以其傳輸速度較快，不保證資料能夠安全的送達，是一個簡單的資料導向通訊協定，由於缺乏可靠性，使用 UDP 應用程式必須能容忍一定數量的封包遺失。例如 video streaming[26]、VoIP 就是典型的 UDP 應用。由於缺乏擁塞控制，當網路發生擁塞時 UDP 還是會用一樣的速度傳送資料，會導致擁塞狀況更加嚴重。

1.2 TCP 簡介

TCP (Transmission Control Protocol)[15]是一種用於IP[13]網路的通訊協定，其特色為連接導向(Connection-Oriented)，資料在傳輸前必須先建立連線，會確保傳送的資料無誤的到達接收端，具有流量控制機制(Flow Control):控制傳送端傳送的速度，擁塞控制機制(Congestion Control):在網路擁塞的時候調整速度避免讓擁塞情形更加嚴重[18]。相較於UDP，TCP確保每個封包無誤到達，當封包遺失TCP會重傳，重傳可能會造成delay time拉長，所以不適用於Time-Sensitive的應用服務，比較適用於像是mail，FTP這類型需要確保資料正確性的應用服務上。

1.3 多媒體影像在 IP 網路中傳輸之議題

Video streaming[26]可容忍有限度的封包遺失。以 MPEG 為例，MPEG[34]是由一張張的畫面(video frame)組成，video frame 分為 I-frame (Intra frame)、P-frame (Predicted frame)與 B-frame (Bidirectional frame)。Video frame 的最小解碼單位為 slice，一個 video frame 是由許多 slice 組成，當每個畫面收不到足夠的 slice 時，video frame 就無法被解碼回來。若只有少數幾個 video frame 無法解碼，人眼將會自動忽略這些小細節，若有很多 video frame 無法解碼，將會有影片播放斷斷續續的情況發生。

由於接收端連續不斷的播放，video frame 必須在限定的時間送達接收端，若超過時限，就算接收端即使收到完整的 video frames，影像也無法播放，會造成影片播放不流暢或影片中斷等現象。

不同的 video frame 對品質有不同程度的影響，I-frame 係指從原影像壓縮之後得來的影像，P-frame 則參考前面的 P-frame 或 I-frame 來組成自己的畫面，B-frame 則透過前後參考 P-frame 或 I-frame 的影像來組成自己的畫面，所以 video frame 在播放與編碼(傳送)的順序是不一樣的，從圖 1.1 可以看到播放順序為 **I、B、B、P、B、B、P、B、B、I**，而編碼(傳送)順序為 **I、P、B、B、P、B、B、I、B、B**，I-frame 的到達是相當重要的。當 video frame 送到底層被切成一個個小封包，在傳送時若能讓屬於 I-frame 的封包全數到達，達成對 I-frame 的完全解碼，此解碼對往後 P-frame、B-frame 的參考相當重要，也因此可提昇整體畫面的品質。

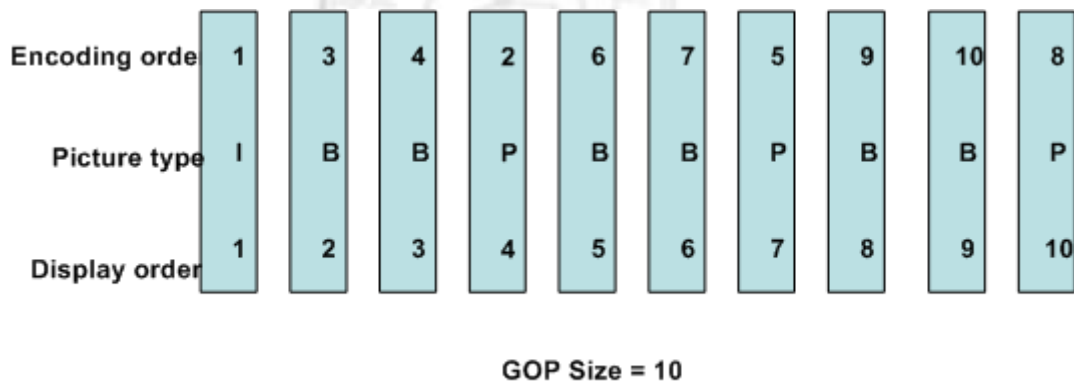


圖 1.1 Encoding Order and Display Order of Frames

1.4 研究動機與目的

如前所述，在新世代網路上，傳統的TCP/UDP兩種傳輸協定不一定能應付層出不窮的新型態服務，在本研究中，我們將改良TCP加入選擇性保護封包機制，讓上層的應用程式決定部分封包不必保證送達。TCP只需依照指示，選擇性的保

證部分封包的傳送，當保證的封包遺失時，本協定會在一定的次數內重傳此封包，藉此，可以確保重要性較高的封包盡可能送達，又不會重傳遺失的不重要封包以節省網路頻寬的消耗，並降低接收端的平均等待時間，我們將以video之傳送為例，證明此法的優越性。

1.5 論文組織架構

本文之第一章分析越來越多的新興資訊服務其資料有重要等級之分，以及其在 IP 網路中傳輸的議題，再分析現今網路中常用的傳輸通訊協定 TCP 及 UDP。評估傳輸多媒體影像這類不須保證全部封包到達的應用服務時時所遭遇到之各種效能議題；第二章介紹一些背景知識以及其他研究學者對於相關議題的研究；第三章介紹本研究所提出之 Partial-Reliable TCP 之細部設計；第四章展示本研究之協定以模擬方法評估我們提出協定的運作效能；第五章為結論與未來展望。

第二章

背景與相關研究

我們將於本章介紹相關的研究以及背景知識。

2.1 TCP 介紹

TCP (Transmission Control Protocol)[15]是現今網際網路上最被廣泛使用的傳輸層協定。它在兩個終端節點之間提供可靠的資料傳輸，現今很多的網路應用程式便是以TCP當作其通訊協定的基礎。

TCP 基本的策略是不斷的送出封包到網路中，並依事件的發生做反應。原始的 TCP 擁有簡單的流量控制機制，但是並沒有擁塞控制的能力。在觀察到一連串的擁塞所造成的網路崩潰後，Jacobson 在 1988 年介紹了一些創新的 TCP 擁塞控制機制[8,9]，從最初的版本 TCP Tahoe[8]到目前最被廣為使用的到目前最常使用的 TCP Reno[7]，和針對其改良的 TCP NewReno[6]等等。包含了慢啟動(Slow Start)[19]、AIMD (Additive Increase and Multiplicative Decrease)[3]還有快速重傳(Fast Retransmit)[21]、快速回復(Fast Recovery)[21]。以下我們介紹 TCP 擁塞控制機制。

2.1.1 擁塞控制機制

大部分的 TCP 是以視窗為基準的傳輸協定，它藉由調整視窗的大小來做流量的控制，利用是否收到 ACK 來判斷是否成功傳輸。其擁塞控制主要是藉由擁塞視窗(Congestion Window)來做速度的控制而非直接以傳送速率來做控制，大部

份 TCP 所採用的方式是藉由封包的遺失或是逾時來做為擁塞的產生與否的判斷並啟動擁塞控制程序。

2.1.1.1 TCP Tahoe and TCP Reno 的擁塞控制

表2.1列出了一些和TCP有關的RFC文件。

表 2.1 TCP 實作相關的 RFC 文件

RFC number	Topic
793	Transmission Control Protocol
1323	TCP Extensions for High Performance
2018	TCP Selective Acknowledgement Options
2581	TCP Congestion Control
2914	Congestion Control Principles
3168	The Addition of Explicit Congestion Notification (ECN) to IP
3390	Increasing TCP's Initial Window
3782	The NewReno Modification to TCP's Fast Recovery Algorithm

TCP Tahoe提出了慢啟動，擁塞避免和快速重傳的觀念。TCP Reno則增加了快速回復的觀念。當在快速重傳階段重傳出遺失的封包時，會進入快速回覆階段，直到收到重傳封包的ACK時，才離開快速回復階段，回到擁塞避免階段。在快速回覆的階段，TCP傳送端和慢啟動階段一樣，每收到一個ACK，擁塞視窗就加 1。

TCP Reno 是以視窗為基礎的擁塞控制機制，其主要分為四個狀態，慢啟動(Slow Start)、擁塞避免(Congestion Avoidance)、快速重傳(Fast Retransmit)、快速

回復(Fast Recovery)，如圖 2.1 所示。

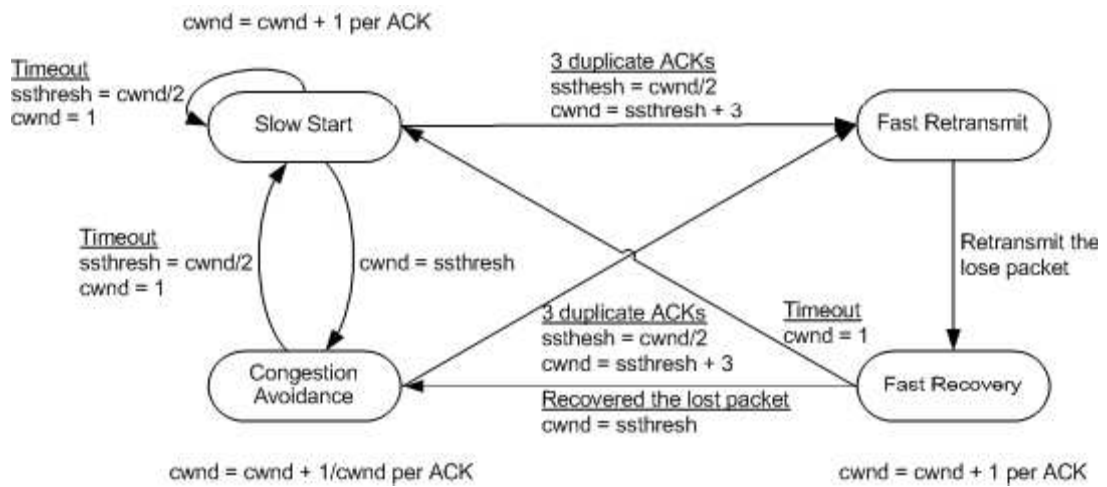


圖 2.1 TCP Reno

2.1.1.2 慢啟動(Slow Start)

在 TCP 連結建立的開始或是逾時產生後啟動慢啟動機制，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 後便增加擁塞視窗之大小，以倍數的方式遞增直到產生封包遺失，當接收端逾時，Congestion Window 會降至初始值，再以倍數成長至 threshold，進入擁塞避免階段。這個方法是希望緩慢的初始速度藉由倍數的成長來加快以達到最適頻寬。

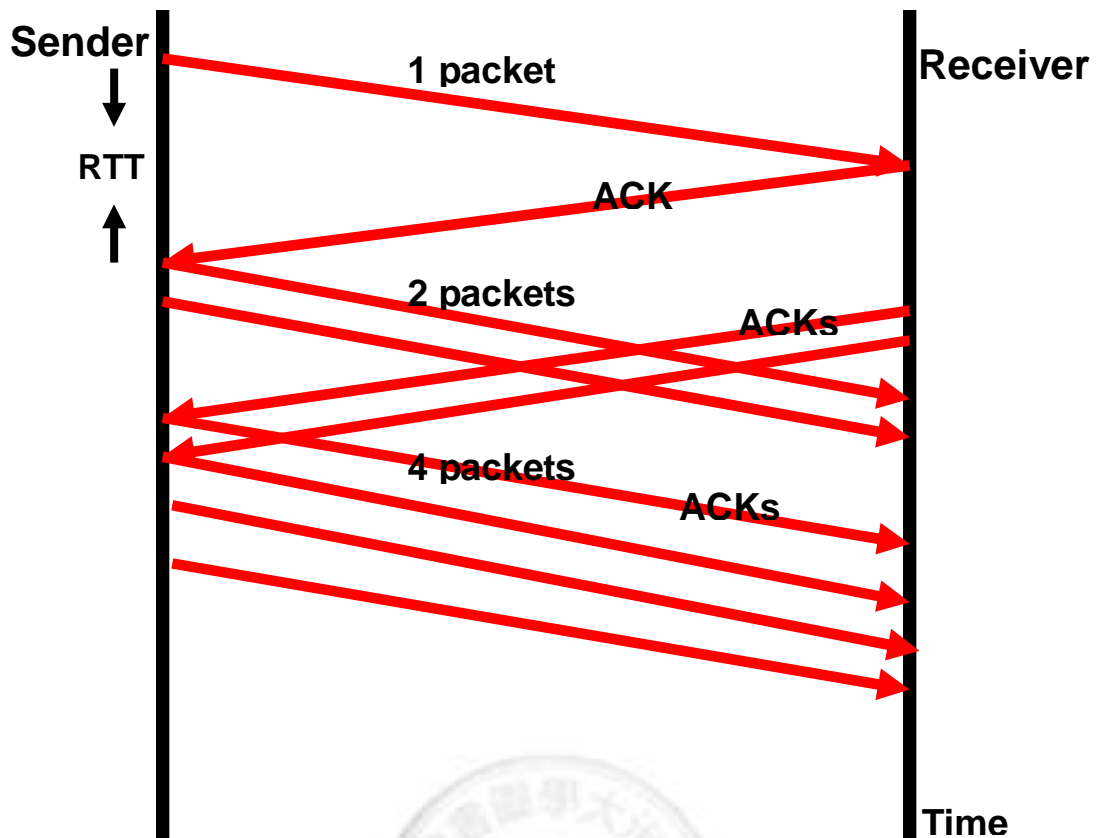


圖 2.2 Packet Flow in Slow Start Phase

2.1.1.3 擁塞避免(Congestion Avoidance)

擁塞避免階段是在擁塞發生之後的速率控制機制，一則必須設法解除擁塞，二則既已偵知速率之極限，理當調整更恰當的速率。不同的 TCP 會做不同的方式，像是 AIMD、Fast Recovery、Fast Recovery 等等。

AIMD 含有兩個階段：

- Additive Increase : 為了避免又產生擁塞，Congestion Window 在每一個 RTT 一次加一個 Maximum Segment Size (MSS)。
- Multiplicative Decrease : 當擁塞產生後，擁塞視窗之大小減半。

當一個 TCP 的連線建立後便進入 Slow Start 階段，這個階段的目的是讓 TCP 藉由不斷的增加 CWND 的大小使得注入網路的資料量越來越大以間接的探索網路的可用頻寬。每接收到一個回覆的 ACK，CWND 便增加一個單位，如圖

2.2 所示，初始的時候，傳送端送出一個 CWND 的封包，當第一個 ACK 接收到以後，CWND 便增加成為 2，當下一個 RTT 送出封包的 ACK 又收到以後，則增加成 4，依此類推，每一個 RTT 之後，若無封包遺失，CWND 大小便以兩倍遞增。當接收端逾時，CWND 會降至初始值，再以倍數成長至 threshold。

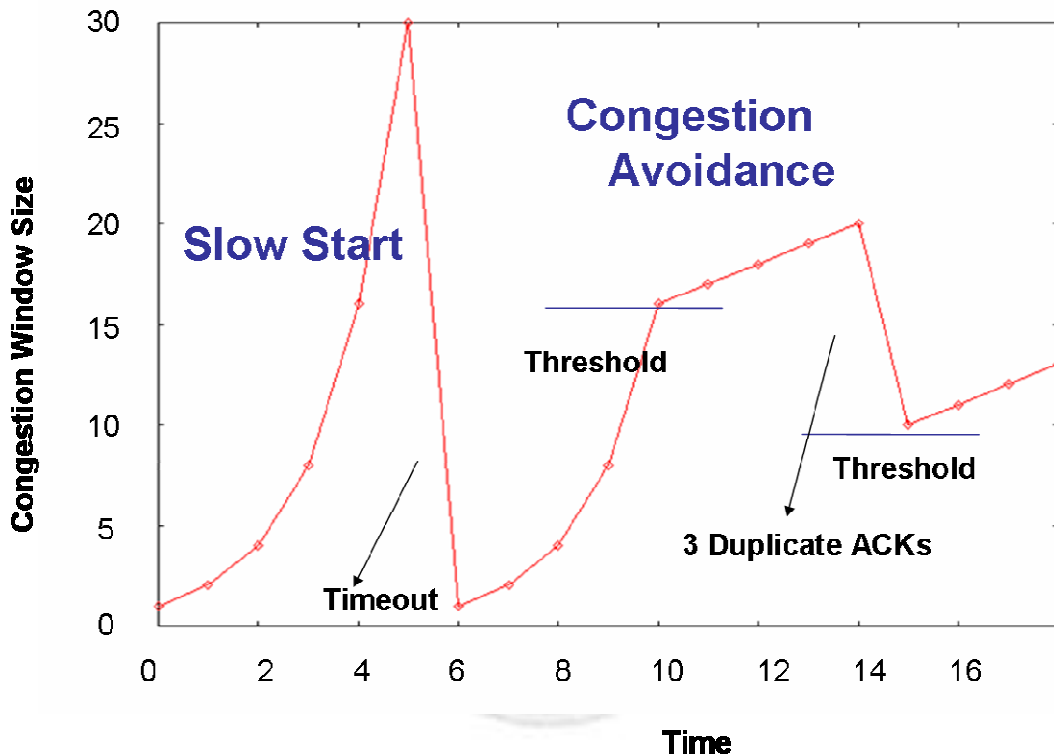


圖 2.3 Change of TCP Congestion Window Size

由於這種快速增加速率的方法，必定會使傳送速率超過網路容量，而導致網路的擁塞發生。在擁塞避免階段，CWND 在每接到一個 ACK 後以線性的方式增加 CWND 的大小，這個過程持續不斷的進行，直到擁塞產生後而 threshold 的大小被減半，如圖 2.3 所示。

TCP Reno 藉由兩種方法判斷封包之遺失，一是 Timeout 的發生，另一個則是接收到三個重覆的 ACKs。當傳送端接到了三個重覆的 ACKs，便開始執行 Fast Recovery 和 Fast Retransmit 演算法。

由於等到 Timeout 發生後再進行重傳常常會耗費過久的時間，造成效能高度

的下滑，因此有了 Fast Retransmit 機制，當傳送端收到三個重複的 ACK 後，便假設此重複 ACK 號碼之下一個封包已經遺失，立即進入重傳的動作。

每當 TCP 接收端收到跳序的封包時，它會送回一個 duplicate ACK 給傳送端，在收到第一及第二個 duplicate ACK 時，接收端會先視為是網路上的延遲，但是如果繼續收到第三個或是更多的 duplicate ACK 時，那麼傳送端就當作此 duplicate ACK 序號之後的封包遺失了，在還沒發生 Timeout 之前，就馬上重送遺失的封包，這就是 Fast Retransmit。

在 Fast Retransmit 之後，所進入的狀態可以是擁塞避免階段，而非慢啟動階段，此機制稱為 Fast Recovery。Fast Recovery 則是基於封包守恆的原則，同一時刻在網路中傳輸的封包數量是恆定的，只有在封包離開網路後，才能發送新的封包進入網路。因此，當 TCP 傳送端收到三個重覆的確認訊息，意謂著緊接在遺失封包之後已經有三個封包成功的到達接收端，也就是網路擁塞情況並不那麼嚴重，因此它不需要把 CWND 大小縮到 1，而大幅度降低傳輸量。所以 Fast Recovery 便是在快速重傳後，將 threshold 設為目前 CWND 的一半然後將 CWND 設為 threshold 加三(因為有三個封包到達接收端)。在這個階段，CWND 增加的方式是以每收到一個 duplicate ACK 時，CWND 的值就加一。如此一來，就不用因此大幅度降低傳輸速度，因此可以提高效能。

2.1.2 資料的封裝

圖 2.4 說明應用程式的資料是如何透過 TCP 在網路上傳輸:

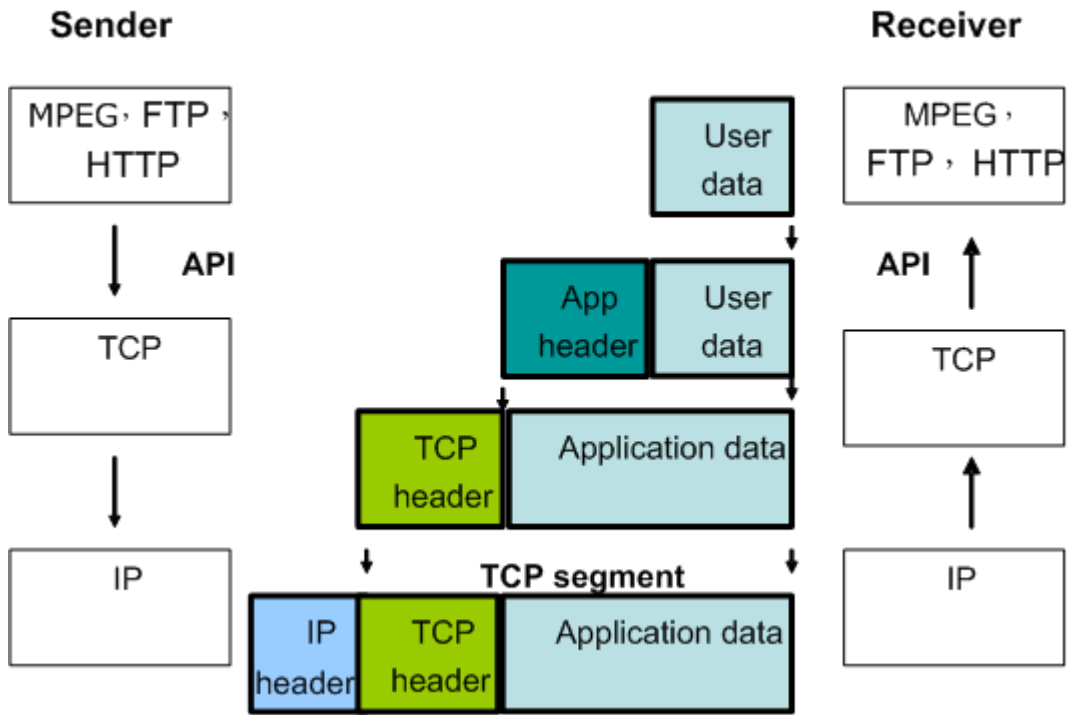


圖 2.4 User Data Forwarded Thru Layers of Protocols

TCP 從應用程式接收資料，並將此資料作為位元組資料流來處理，大多數網路因為硬體設計的限制，所能通過的資料大小通常只有數千 byte 而已，許多需要在網路中傳輸的資料都會超過這個數值，所以 TCP 必須將這些位元組分段切成一個個封包，再將這些封包進行編號及安排傳遞的順序，並將這些封包正確的傳到接收端，最後將封包裡的位元組資料流重組還原成應用程式的資料。

2.2 Video streaming

Video streaming[26]是一種具有即時特性的傳輸技術，相較於傳統收看存放在網路上的影音資料必須等到全部下載到用戶端設備後才可開始播放的方法，串流技術讓用戶端只需預先下載極小部份影音資料即可開始播放，大量節省了使用者在等待下載的時間，例如線上教學、視訊會議、線上廣告、隨選視訊、即時線上轉播以及網路電台等都是video streaming的應用。

2.2.1 Structure of video streaming

Video streaming大致組成元件及流程如圖2.1所示，首先編碼器將媒體擷取裝置產生的多媒體影音資料編碼後或是直接取出其他儲存的數位化多媒體資料，產生可以支援video streaming格式的封包流，經由伺服器透過網際網路將資料傳送到用戶端，用戶端經過解碼程序後即可播放。

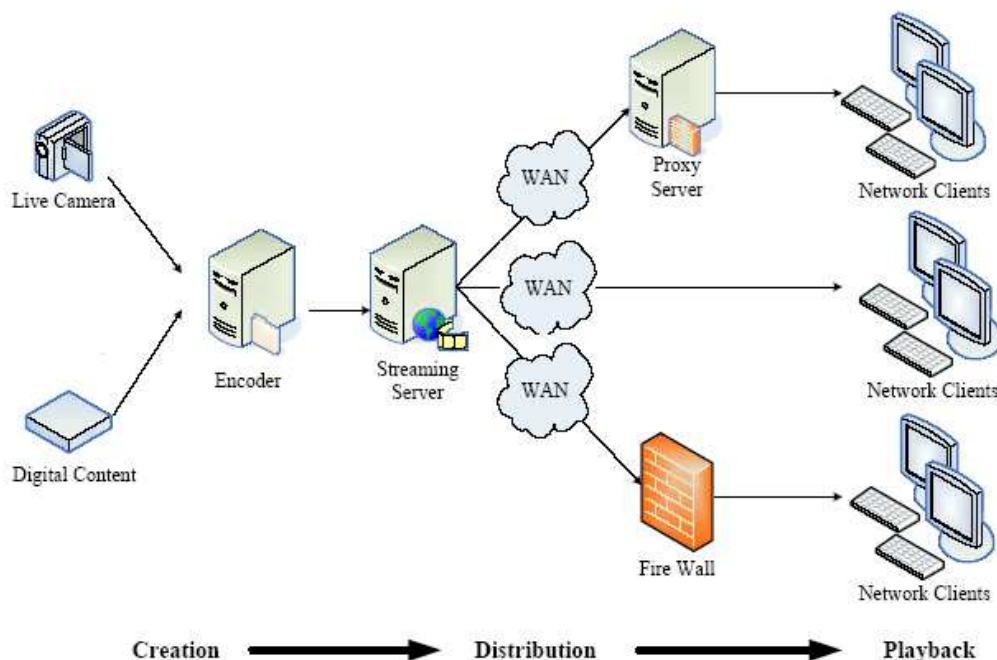


圖 2.5 Video streaming over IP network

2.2.2 影像壓縮技術

Video的資料量通常都很大，要在網路上傳輸那麼大的資料量是非常耗費頻寬的，通常video資料都會先經過壓縮編碼，再放到網路上傳輸。視訊壓縮有很多不同的壓縮標準，適用於不同的應用範圍，MPEG (Motion Picture Experts Group)[34]算是最普遍地被應用於數位視訊服務的一個視訊壓縮標準，下節將介紹MPEG。

2.2.3 MPEG (Motion Picture Experts Group)簡介

MPEG 是動態視訊專家小組的簡稱，ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) 在 1988 年[34]對動態視訊與語音壓縮標準提出的壓縮準則。連續的 video frame 之間的相似度很大，而一個 frame 內相鄰的 pixel 值通常會比較接近，影像壓縮的目的就是將相似重複的資訊去除，先計算相鄰圖片的差異，再將這些差異做處理。為了取得高壓縮比效果，MPEG 採用了複合式多種壓縮技巧，首先是以區塊為基礎的動態補償方法，利用前一畫面至目前畫面內容之預測，或是由前一畫面其下移畫面至目前畫面內容之內插預測，其預測的差異值，再利用離散餘弦轉換(Discrete Cosine Transform, DCT)[28]除去空間上的相關性，並配合量化(Quantized)[28]程序略除不重要的資訊，最後經由 VLC (Variable Length Coding)[28]方式編碼後與動態向量複合產生視訊壓縮編碼。

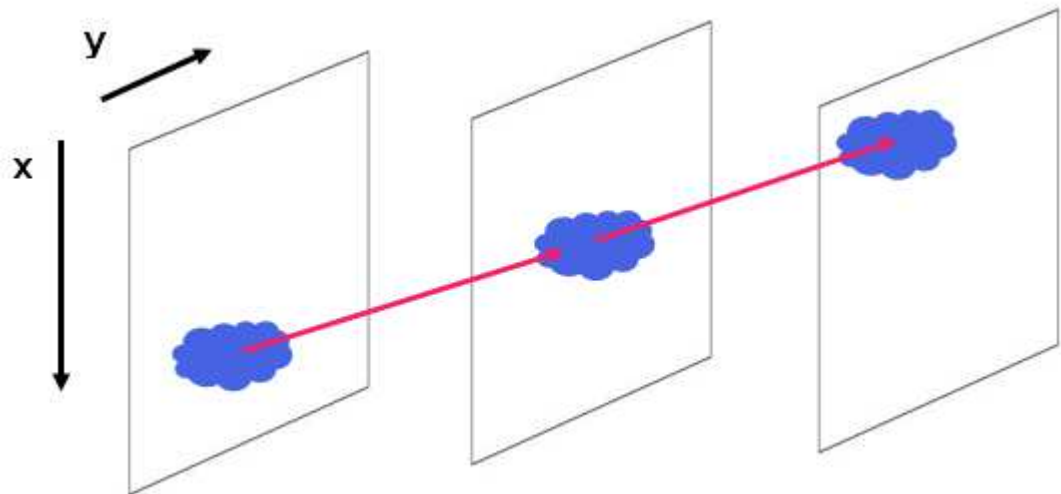


圖 2.6 Block of Frames in MPEG

隨著年代的不同 MPEG 壓縮技術又延伸出不同的版本，MPEG-1、MPEG-2、MPEG-4，...，不同版本適用於不同的環境。

MPEG-1:

MPEG-1[34]制定於 1992 年，可適用於不同容量的設備，如 CD-ROM、Video-CD、CD-i，傳輸速率為 1.5Mbits/sec，每秒播放 30 張，具有 CD 音質，畫質基本上與 VHS 相當。MPEG-1 的編碼速率最高可達 4-5Mbits/sec，但隨著速率的提高，其解碼後的圖像品質有所降低。MPEG-1 也被用於網路上的視訊傳輸，如隨選視訊系統。

MPEG2:

MPEG-2[35]制定於 1994 年，設計目標是更高的傳輸率，其所能提供的傳輸率在 3-10Mbits/sec 間，其解析度可達 720x486。MPEG-2 的音訊編碼可提供左右中及兩個環繞聲道，以及一個加重低音聲道，和多達 7 個伴音聲道(DVD 可有 8 種語言配音的原因)。MPEG-2 提供一個較廣的範圍改變壓縮比，以適應不同畫面品質、儲存容量，以及頻寬的要求。因為現存電視機解析度限制，MPEG-2 所帶來的高清晰度畫面質量在電視上效果並不明顯，倒是其音訊特性(如加重低音、多伴音聲道等)更引人注目。

MPEG-4:

MPEG-4[25]於是針對一特定比率下的視訊、音訊編碼，注重多媒體系統的交互性和靈活性。MPEG-4 標準主要應用於視訊電話、視訊電子郵件和電子新聞等，其傳輸速率要求較低，在 4800-64000bits/sec 之間，解析度為 176x144。MPEG-4 利用很窄的頻寬，通過重建技術，壓縮和傳輸資料，以最少的數據獲得最佳的圖像品質。與 MPEG-1 和 MPEG-2 相比，MPEG-4 更適於在網路上傳輸。下節我們將簡介 MPEG 的壓縮原理。

2.2.4 MPEG 壓縮原理

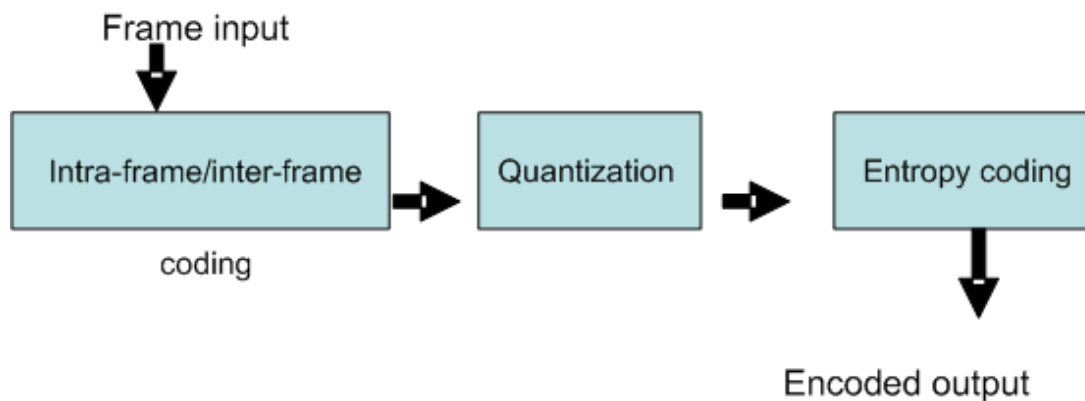


圖 2.7 Encoding Percedure of MPEG

上圖是 MPEG 視訊編碼的基本流程，主要有三個步驟，其中最重要的部份是畫面內編碼(Intra-frame coding)[34]或畫面間編碼 (Inter-frame coding)[34]。

畫面內編碼 (Intra-frame coding):

MPEG 視訊壓縮標準定義將每個畫面分割成 8*8 像素大小的數個區塊，以區塊做為基本的像素轉換編碼單位，其目的為降低處理程序的延遲及複雜性。為了降低像素間的相關性，每個區塊個別做離散餘弦轉換(DCT)。將空間信號轉換成頻率信號，轉換後信號能量大部份集中在少數的低頻係數上，只有少數部份能量會分

布在高頻係數上。隨後的量化程序，能方便及有效地達成資料壓縮的目的，而且人類的視覺系統對高頻訊號改變的敏感度較小，對於低頻訊號的改變反而比較容易辨別，所以對於不同頻率的量化位階大小的選擇，也是視訊編碼的重要關鍵之一。

畫面間編碼 (Inter-frame coding):

移動估測(Motion Estimation)以及移動補償(Motion Compensation)：移動估測的目的在於畫面編碼時，到周圍的參考畫面中尋找與目前畫面相似的區域，找到之後並記錄此區域移動了多少向量，稱移動向量(Motion Vector)；而移動補償則是反過來利用已知的移動向量和鄰近參考畫面，將正確的影像還原回來，一般是以大區塊作為移動估測與移動補償的基本單位。在連續的畫面間，相同的位置周圍的像素值有很高的相關性，畫面間編碼主要在消除時間上像素的相關性，達到降低位元率的目的。如圖 2.8，畫面分割成數個大區塊，欲編碼的畫面 frame n 會去前一張或後一張重建的參考畫面 frame n-1 的固定搜尋區域內做區塊比對，假始能在重建畫面 frame n-1 中找到一個很相似的大區塊，則將大區塊間的移動向量進行編碼，並且將兩個大區塊間的差異值傳送到量化程序處理，若找不到相似的大區塊，則直接送到量化程序處理。

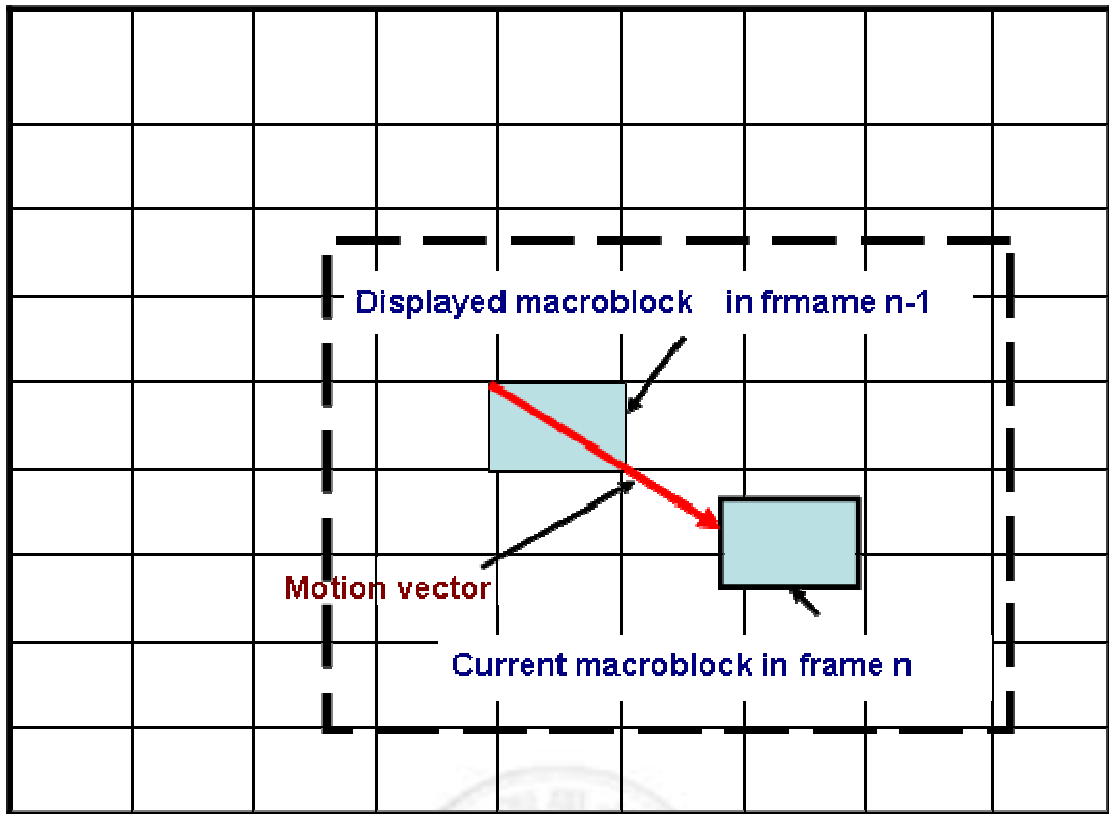


圖 2.8 Motion Estimation of MPEG

MPEG 定義了三種畫面的編碼模式[34]，分別為 I-frame、P-frame 與 B-frame:

I-frame (Intra-coded pictures) :

I-frame 僅使用本身的資料進行編碼經量化後再經 VLC (Variable Length Coding) 編碼，沒有參考其他畫面的資料，所以在解碼時不需參考其他畫面的資料，可以單獨進行解碼。I-frame 是畫面群組的第一張，隨後在畫面群組裡的 P-frame 與 B-frame 都會參考到它的資料，壓縮率低。

P-frame (Predictive coded pictures) :

P-frame 在解碼時，會使用到參考畫面的資料，這些參考畫面為前面較早被播放的 I-frame 或 P-frame，而參考的位置就是以移動估測所產生的移動向量來表示，若找不到最適合的大區塊時，則使用 Intra 模式編碼。由於參考前一 I-frame 或 P-frame 且以動態補償方式預測編碼，其壓縮效率較高。

B-frame (Bidirectionally predicted pictures) :

B-frame 在解碼時，會使用到前面及後面兩個方向參考畫面的資料。如同 P-frame

一樣，畫面資訊在參考畫面找不到相似的大區塊時，會使用 Intra 模式編碼。參考前後畫面做動態補償預測編碼，擁有最高的壓縮率，本身不再做為其他預測編碼用。

如圖 2.9 所示，我們列出一畫面群組(包含 I-frame、P-frame 及 B-frame)，第一畫面 I 為獨立編碼，它與其它畫面沒有任何關連。第四張畫面 I 是參考第一張畫面 I，此種編碼方式為順向預測(Forward Prediction)。而第二張畫面 B 不僅參考了第一張畫面 I 還包括了未來第四張畫面 P，此種作法稱為內插預測(Interpolation Prediction) 的方法。消除畫面間時間和空間的關聯性，僅用最小的資料量來記錄彼此間的差異，也就是利用 Motion Vector 來記錄，如此可大大的增加壓縮的效率。編碼的順序和播放的順序是不同的，編碼的順序是 I => P =>B，播放的順序則是 I =>B =>P。

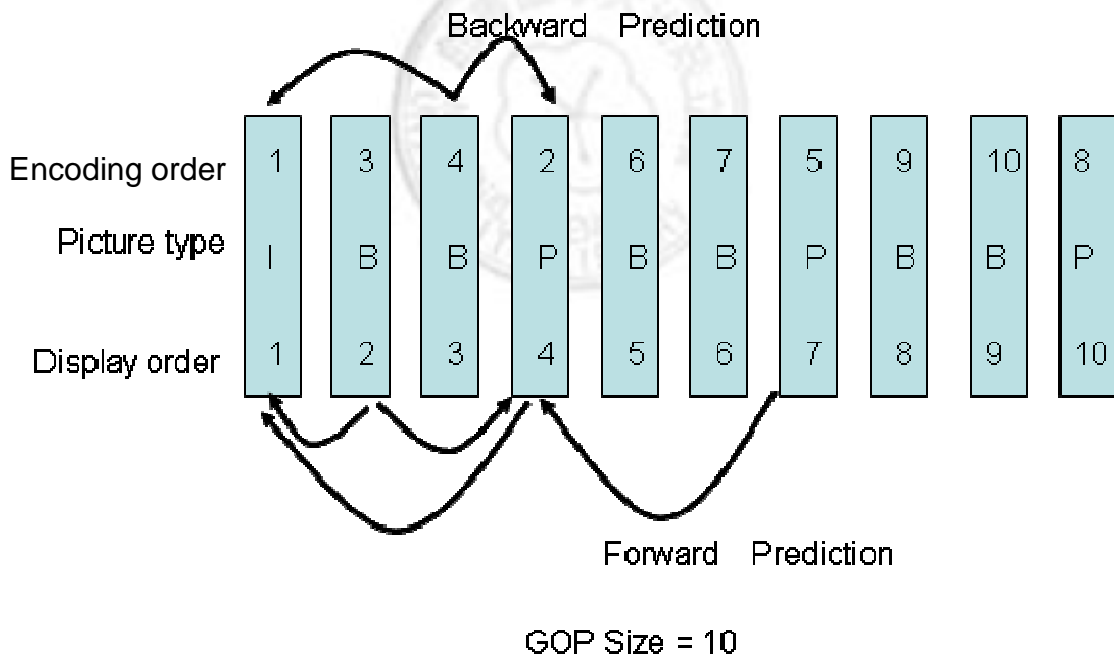


圖 2.9 Order of Edcode and Display per GOP

2.3 相關研究

目前提出改善 video streaming 這類應用服務的相關研究非常多，我們將於本節介紹。

H.264/AVC:

H.264/AVC (Advanced Video Codec)[27]是新一代的影像編碼技術，於 2003 年被提出，目前最新的是 2005 年 12 月發表的第三版。H.264 又稱為 MPEG 4 Part10 或 AVC，主要是由 JVT (Joint Video Team)所提出的規格，其壓縮率及視訊品質皆優於 MPEG 2/4，與其它影像編碼主要不同之處在於它包含了兩個 layer，分別是 VCL (Video Coding Layer)與 NAL (Network Apstraction Layer)。其特色在於可以將其最小的解碼單位亦即一個 slice 單靠本身的壓縮資料就能解碼，而不必依靠其他 slice，優點是當傳送到遠端時，每接收完一筆 slice 的壓縮資料就能馬上解碼，不用等到整張資料接收完後才開始，且萬一傳送過程中發生資料遺失或錯誤，影響所及也僅限於該筆 slice，不會對其他 slice 有所影響。一般來說，在傳輸 H.264/AVC 影像封包時，在壓縮時就會設定一個 slice 編碼構成一個封包，使其不會因為網路環境的影響造成無法解碼的狀況。當遇到頻寬改變時，server 會送出適合當時頻寬大小的 bit-stream，讓影像比較平順。

SRP:

SRP (Selective Retransmission Protocol)[17]是應用層上的協定，使用 UDP 作為傳輸層的通訊協定，當封包遺失時，選擇性重傳一定比例的封包，SRP 會根據目前的封包遺失狀況，延遲的時間，以及 RTT 的時間來決定要重傳多少比例的遺失封包。

SRP 要估算封包遺失狀況、延遲時間以及 RTT 的時間，這樣的 overhead 太大，且只重傳一定比例的遺失封包，若這些被重傳的封包中是比較不重要的封包，必會造成頻寬資源浪費。

SCTP:

SCTP (Stream Control Transmission Protocol)[1]是傳輸層通訊協定，它提供了 Partial Reliability 模式，未啟動PR 模式時，就如同TCP一樣提供可靠的傳輸服務，若啟動則提供個部份可靠的傳輸模式，在傳送封包時可以指定此一封包之生命時間，若在時間內無法將封包成功傳送，SCTP 便會放棄該封包的傳送。

該做法與我們的研究相近，但是SCTP並不普遍，不像Partial-Reliable TCP比較容易推廣實行。

DCCP:

DCCP (Datagram Congestion Control Protocol)[10]是傳輸層通訊協定，簡單來說，DCCP就是有ACK的UDP，利用ACK所得到的資訊去做Congestion Control，較不在乎資料完整性，讓已經在使用的訊務服務達到傳輸時效性，為其主要目的。

DCCP 雖然可以確保部份封包到達，但是在重傳遺失封包時，沒有彈性，也可能會造成傳輸時間過長以及頻寬資源浪費，且不像 Partial-Reliable TCP 比較容易推廣實行。

TFRC:

TFRC (TCP-Friendly Rate Control)[7]是傳輸層通訊協定，別於 TCP 使用 AIMD，TFRC 使用 AISD (Additive Increases and Subtractive Decreases)控制傳輸速率機制，在反應網路擁塞時可以有較小的流量變化，也因為這樣的特性使得此機制適用於需要平順傳輸率的串流媒體應用。

但是 TFRC 無法根據封包重要等級不同做不同應變，無論重要或是不重要封包遺失時都會重傳，可能會造成傳輸時間過長以及頻寬資源浪費。

在2004年Vieron等人[21]根據TFRC的觀念，提出了一種考慮即時限制 (Real-time Constrained)的視訊位元率控制機制，但是根據TFRC來對視訊作位元率控制將面臨一個問題，即在可用頻寬劇烈變動的情況下視訊品質也將產生變動。有些研究對視訊訊務的統計分佈作分析，並嘗試以訊務模型(Traffic Model)來描述視訊訊務[4,5]，然而隨著壓縮編碼標準的發展和演進，舊的訊務模型其訊

務描述的能力將會受到限制，甚至於完全不能符合實際的情況。在各種利用已知的訊務特性作平順化的演算法方面[6]，主要的目標即盡量消除視訊的叢集特性以降低峰值頻寬需求(Peak Bandwidth Requirement)，1997年Feng等人提出最少傳輸率變化次數的平順化演算法(Smoothing Algorithm)[7]，接著在1998年Salehi[19]等人則提出最小傳輸率變化的最佳化平順化演算法，在考慮到用戶端記憶體大小的條件下，以離線的方式最佳化平順傳輸規劃。2000年Sen[20]等人利用平順視窗(Window)的觀念，在平順視窗內對有限的已知視訊訊務使用離線平順化演算法，作平順傳輸規劃的計算，並可保證伺服器端及用戶端的記憶體皆不會發生溢滿(Overflow)的情形。由於平順化傳輸規劃會受到網路延遲變動和可用頻寬變動的影響，可能無法符合原先規劃而降低接收視訊的品質。

無線網路上的改良:

目前也有相當多研究是針對多媒體在無線網路中傳輸的議題，[23]中提到了一些跨層設計的方法，Top-down、Bottom-up、Application-centric、MAC-centric 與 Integrated 等方法。

Top-down approach: APP layer 最佳化參數與策略，由跨層方式傳至 MAC layer，再由 MAC layer 決定 PHY layer 的 modulation。

Bottom-up approach: 由底層自行決定頻寬變化與損失，上層無法確知下層決策。

Application-centric approach: 由 APP layer 最佳化底層傳上來的參數，因應 application 使用需求，決定使用 top-down 或 bottom-up 的方式。

MAC-centric approach: APP layer 將 traffic information 與需求傳給 MAC layer，由 MAC layer 決定傳輸的 QoS level。

Integrated approach: 整合式設計的作法是跨層設計最完善的作法，將各種策略及計算方法整合在一起，透過跨層最佳化計算器收集所有 layer 的參數。但整合式作法的計算複雜度非常高。

雖然跨層的設計方法可以有效的提升服務品質，但是整體的 overhead 太高，

不容易推廣實行。

2.4 小結

Video streaming 這類資料量大且不用完全保證資料到達的應用服務，如只求傳輸速度是沒辦法改善其服務品質的，網路擁塞的狀況是時常發生的，所以有好的擁塞控制機制是有效減緩網路擁塞所帶來的影響，若又能盡量確保該類服務中較重要的資料到達，一定可以有效的提升該應用服務的品質，在下一章中我們將提出基於以上考量的通訊協定設計。



第三章

Partial-Reliable TCP

我們將於本章中講述我們的方法。

3.1 問題分析

Video streaming 這類資料量極大的應用服務，在頻寬不充裕的 IP 網路上傳輸時不易維持服務品質，雖然現在的網路頻寬不斷的在增加，但是相對的，網路上面的訊務量也越來越多，無論如何的增加頻寬，始終無法讓該類型應用的服務品質得到保證。

當網路資源不足時，必然發生網路擁塞現象，我們在第一章曾提到，UDP 因為沒有擁塞控制機制，在網路擁塞時，仍然維持相同的傳輸速度，導致擁塞況更嚴重。Video streaming 這類型服務的封包的傳送都有時限，逾時的封包等同於無效封包，但是傳輸層的 TCP 卻不理會此封包是否有時限，當網路資源不足時，某些傳送中的封包會被網路中的 router 丟棄，TCP 發現封包遺失便會重傳該遺失封包直到送達為止，超過時限送達的封包會被捨棄，造成資源浪費，且這些遺失的封包如果是比較重要的封包(如 I-frame)，更會嚴重的影響服務品質。

我們嘗試只對部份封包(如 I-frame)做保護，希望在網路資源不足的環境下，當封包遺失時，只針對要保護的封包重傳，降低重傳的 overhead，減少無效資料的接收及傳送量，降低封包遺失對服務品質造成的影響。

3.2 設計理念

本文提出 Partial-Reliable TCP，設定不同的保護等級，讓上層的應用程式可以彈性的決定要使用的保護級別，Partial-Reliable TCP 只需依照指示，選擇性的保護封包，在網路資源有限的情況下維持服務的品質。

Partial-Reliable TCP (PR-TCP)是以 TCP 為基礎作改良，分成以下幾個部份，(1)依照上層的指示，設定封包的保護等級，(2)根據每個封包所設定的保護等級在不同的情況下做不同的處理。我們提出了 Basic PR-TCP，但這版本須更改傳送、接收兩端的通訊協定，雖然會有較好的效能改善但較難推行，所以我們又提出 Single-Side 的版本，只需更改傳送端，效能改善幅度沒有 Basic PR-TCP 那麼大，但較易推廣。我們將於下節介紹我們的方法。

3.3 Basic PR-TCP

Basic PR-TCP 是運行在傳送及接收兩端的通訊協定，在這節中，我們會講述 Basic PR-TCP 的運作機制及流程。

3.3.1 Protection Class

Basic 版本定義三種封包保護等級：

Regular: 一般性的封包，不提供遞送保證。

Certified: 有生命期的重要封包，僅在生命期內提供遞送保護，若逾時則不提供保證。

Registered: 重要的封包，提供遞送保證。

我們將保護等級記錄於封包的 header，新增欄位及說明如下：

pt (packet type) :記錄封包的保護等級。

Bpt (Before packet type):記錄前一個封包的保護等級。

Npt (Next packet type):記錄下一個封包的保護等級。

Basic PR-TCP 從應用程式接收到資料並將這些資料切割成封包時會依照應用程式指示將封包的保護等級填入 header 的欄位中，並且維護一個封包生命期限記錄表，紀錄每個封包的生命期限，Certified class 的封包設定為非零值，

Registered 及 Regular class 設定為 null。

3.3.2 Packet Life Control Scheme for Certified Packets

Packet Life Control Scheme 是用來控制 Certified 封包的重傳時限，當封包重傳的時間超出預設值時，傳送端會發送訊息告知接收端，不再重傳。像 video streaming 這類應用服務對於時間是非常敏感的，過遲的封包並無利用價值，所以 PR-TCP 利用 Packet Life Control Scheme，在時限內重傳 Certified 封包，過了時限則放棄重傳，避免資源浪費。

我們在封包的 header 新增 RL (Retransmission Limit)欄位記錄封包重傳次數限制。新增 FNP (Forward Next Packet) message 告知接收端，不再重傳封包。當封包遺失時，傳送端會在 RL 設定的次數內進行重傳，每重傳一次封包，RL 值就減一，若 RL 的值為零，則發出 FNP 告知接收端放棄重傳，中止等待，接收端不會無限期等待下去。

3.3.3 Selective Retransmission Scheme

Basic PR-TCP 延伸 TCP Sack[16]的很多機制，加入 TCP Sack 的選項允許接收回傳目前已經連續收到的區段，藉此傳送端可根據這些資訊得知那些封包是沒被收到的。表 3.1 為發生封包遺失的時候傳送端及接收端針對不同 Protection Class 所做的處理：

表 3.1 Packet Loss Handling Procedure in Basic PR-TCP

Protection Class	Sender	Receiver
Regular	do not retransmit	do not wait
Certified	if (RL!=0) { retransmit ; RL --}	wait until FNP message is received
Registered	retransmit	wait

以 video streaming 為例，逾時的資料對影像的播放是無效的，在網路資源不足的情形下，必須要盡量確保像 I-frame 這類封包的傳達，避免讓服務品質嚴重受損，我們就可以將 I-frame 設定為 Certified class，而 P-frame、B-frame 對於 MPEG video 而言較不重要，我們可以將其定設定為 Regular class，封包遺失時不重傳，可降低平均傳送時間。

在 2.1.2 節中提到，TCP 如何將上層的位元組資料切割傳送，並在接收端將封包重組給上層，Basic PR-TCP 的接收端會將忽略的遺失封包裡的每個資料位元以”1”替代，交給上層應用程式處理，避免在資料重組階段發生錯誤。

3.3.4 Basic PR-TCP 的擁塞控制機制

Basic PR-TCP 的擁塞控制機制與 TCP 相似，其主要分為三個狀態，Slow Start、Congestion Avoidance、Fast Selective Retransmit and Fast Recovery。

慢啟動 (Slow Start):

Basic PR-TCP 的 Slow Start 階段跟 TCP 的相同，我們不在此贅述。

擁塞避免 (Congestion Avoidance):

當一個 Basic PR-TCP 的連線建立後便進入 Slow Start 階段，速度持續增加直

到 Timeout 發生進入擁塞避免階段，在此階段，接收端的 CWND 在每個 RTT 會以倍數成長至 threshold，接下來便以線性成長，直到發生封包遺失時便進入 Fast Selective Retransmit 和 Fast Recovery 階段。

快速選擇重傳與快速回復 (Fast Selective Retransmit and Fast Recovery):

跟 TCP 不同的是，傳送端可以直接從接收端的 ACK 得知哪些封包遺失，根據遺失封包的 Protection Class 做處理，CWND 的值會減半，再以線性方式增加 CWND。如圖 3.1 所示。圖 3.2 為 Basic PR-TCP 的 state diagram、表 3.2 為每個事件觸發的狀態轉換以及接收端的 CWND 值變化情形:

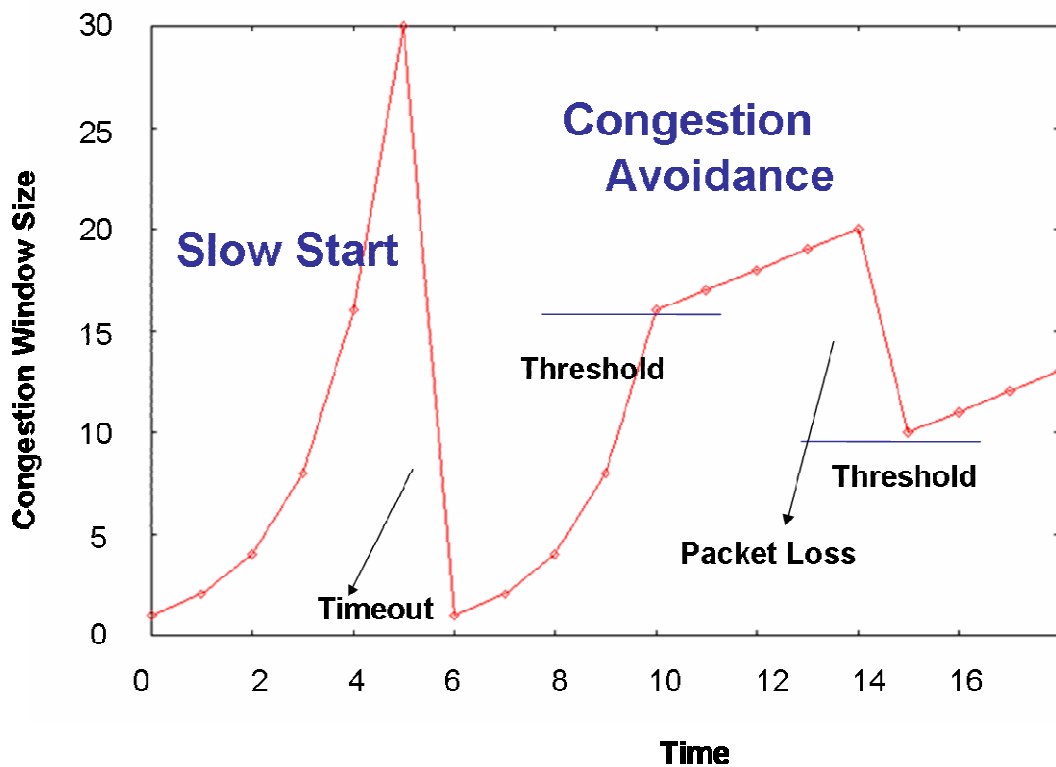


圖 3.1 Change of Basic PR-TCP Congestion Window Size

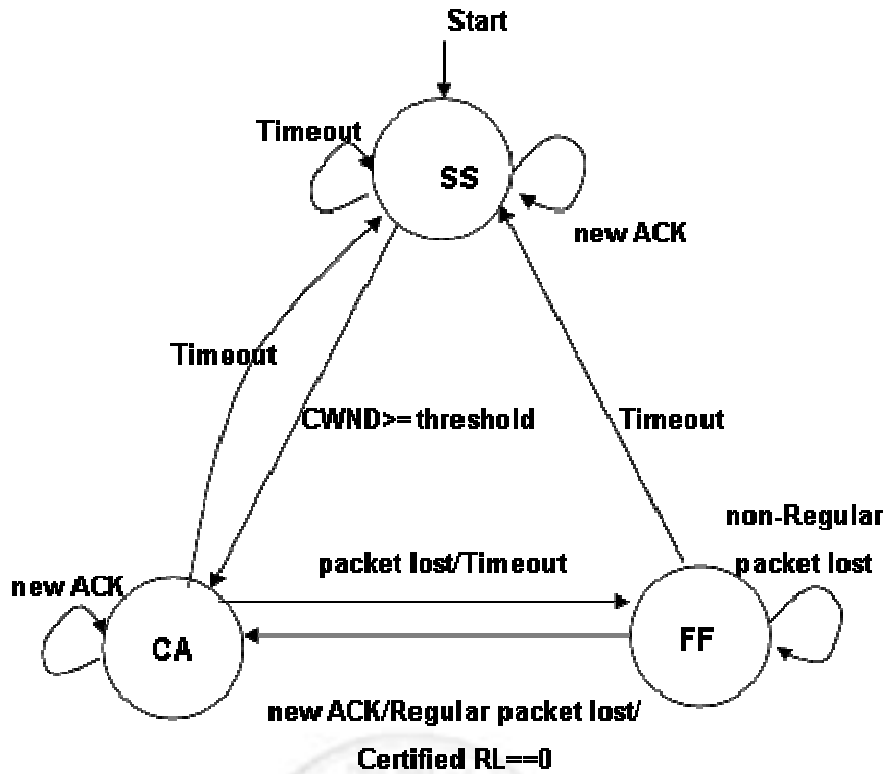


圖 3.2 State Diagram of Basic PR-TCP

表 3.2 State Transition in Basic PR-TCP

Event	Current State	Next State	Action
new ACK	(SS)	(SS)	$CWND = CWND * 2$
Timeout	(SS)	(SS)	$\text{threshold} = (1/2) * CWND$ $CWND = 1$
$CWND \geq \text{threshold}$	(SS)	(CA)	$CWND = CWND + 1$
new ACK	(CA)	(CA)	$CWND = CWND + 1$
Timeout	(CA)	(SS)	$CWND = 1$
packet lost/Timeout	(CA)	(FF)	$\text{threshold} = CWND * (1/2)$ $CWND = \text{threshold}$
non-Regular packet lost	(FF)	(FF)	No change
new ACK/Regular packet lost/ Certified RL==0	(FF)	(CA)	$CWND = CWND + 1$
Timeout	(FF)	(SS)	$CWND = 1$

藉由 Basic PR-TCP 的擁塞控制機制，可以有效的根據網路的狀況調節傳送速度，不致加重網路擁塞狀況。圖 3.3 為 Basic PR-TCP 接收端接收到 ACK 的流程，圖 3.4 為其演算法虛擬碼。



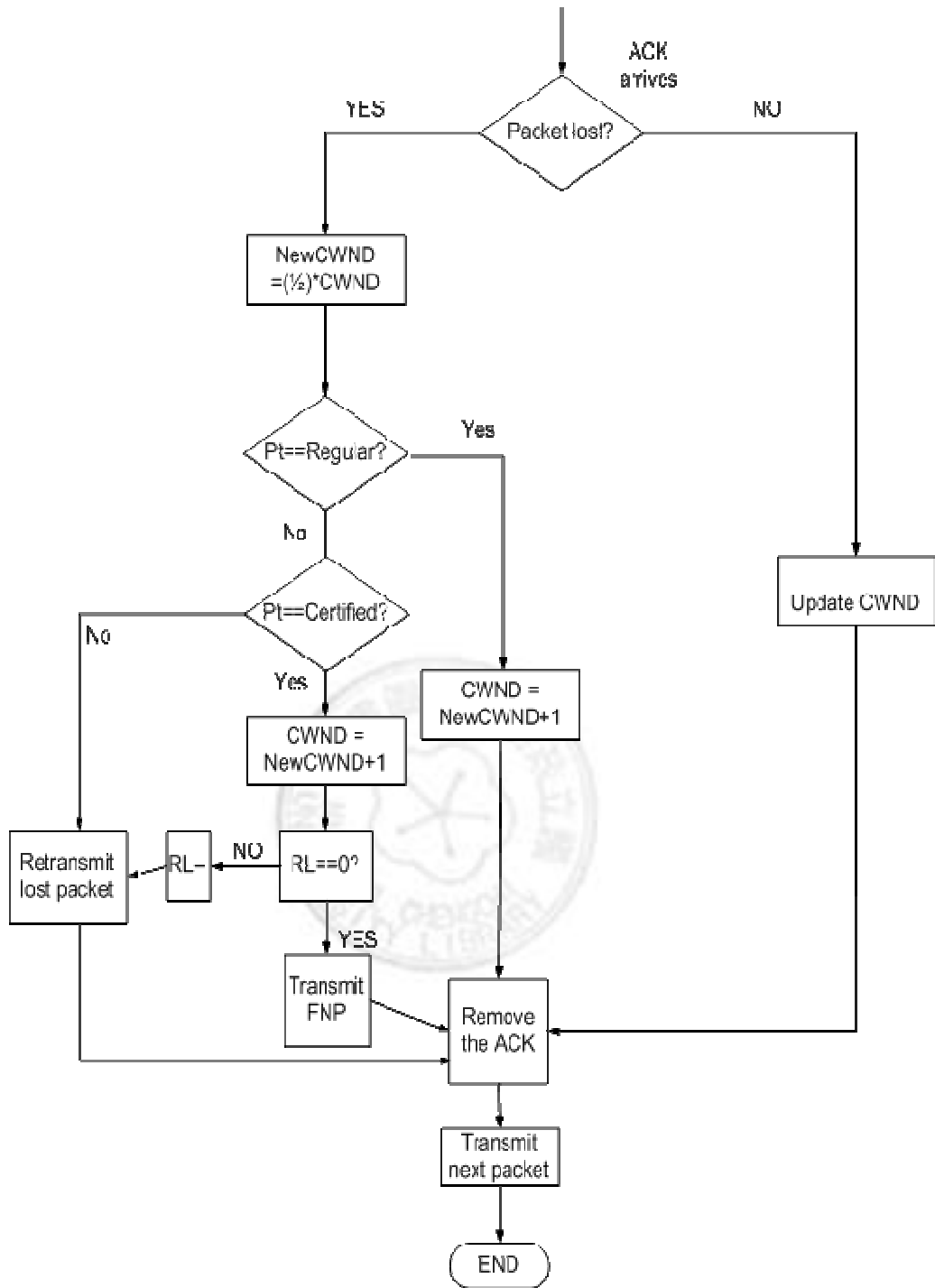


圖 3.3 ACK Arrival Procedure in Basic PR-TCP

```

ACKarrival( ){
    If(packet lost){
        NewCWND=(1/2)*CWND;
        If(Pt==Regular){
            CWND=NewCWND+1;
        }
        Else if(Pt==Certified){
            CWND=NewCWND+1;
            If(RL==0){
                TransmitFNP( );
            }
            Else{
                RL--;
                RetransmitLostPacket( );
            }
        }
        Else{
            RetransmitLostPacket( );
        }
    }
    Else{
        UpdateCWND( );
    }
    RemoveACK( );
    TransmitNextPacket( );
}

```

圖 3.4 Pseudo Code for ACK Arrival Procedure in Basic PR-TCP

3.3.5 小結

Basic PR-TCP 可以根據上層的需求，將封包設定為不同的保護等級，並在封包遺失時，依照不同的保護等級做不同的處理，彈性更高，適用於 video streaming 這類應用服務，但是由於必須更改傳送端及接收端，推行起來會比較不容易。我們將於下節提出只需改變接收端的版本，Single-Side PR-TCP。

3.4 Single-Side PR-TCP

因為使用 Basic PR-TCP 必須在傳送端及接收端都裝上 PR-TCP，比較不容易推行，所以我們另外提出一個版本，Single-Side PR-TCP，只需裝設在傳送端。在 client/server 這種一對多架構中，只需在 server 端裝設，而 client 端可以使用任何 TCP，如此推行 PR-TCP 較為容易。

由於只更改傳送端的傳輸協定，接收端仍使用其他 TCP，若有部分的封包遺失，而傳送端並未重送時，接收端就會無限期的等待下去，直到封包重傳成功。Single-Side PR-TCP 必須有機制解決這個問題。

3.4.1 Protection Class

Single-Side 版本定義了兩種保護等級：

Regular: 一般性的封包，不提供遞送保證。

Registered: 重要的封包，提供遞送保證。

跟 Basic PR-TCP 不同的地方在於 Single-Side PR-TCP 沒有 Certified Class，原因在於 Single-Side PR-TCP 僅運行在傳送端，接收端仍是其他的 TCP，無法配合傳送端做相對應的處理。

3.4.2 Replication Scheme

前面提及，若有封包遺失而沒有重送時，接收端的 TCP 會無限期等待下去，為了解決這個問題，我們讓傳送端複製 Regular 封包的 header，直接重複送出。一般的 TCP 封包大小最大為 1500byte，其中 header 大小為 20byte，接收端在收到重複編號的封包時會直接丟棄，複製多個只有 header 沒有 payload 的假封包傳送，並不會影響接收端正常運作。假封包可以降低接收端因為封包遺失而一直等待的機會。

我們藉由實驗評估適合的假封包數量，設定實驗環境為較容易發生封包遺失的網路環境，傳輸 10MB 的資料，每個傳送的封包都複製假封包，並控制網路中的封包遺失率，觀察 100 次的實驗中在不同封包遺失率情形下，假封包數量對 Timeout 發生的影響以及當發生 Timeout 時的 overhead 估算。實驗結果如圖 3.5 所示：

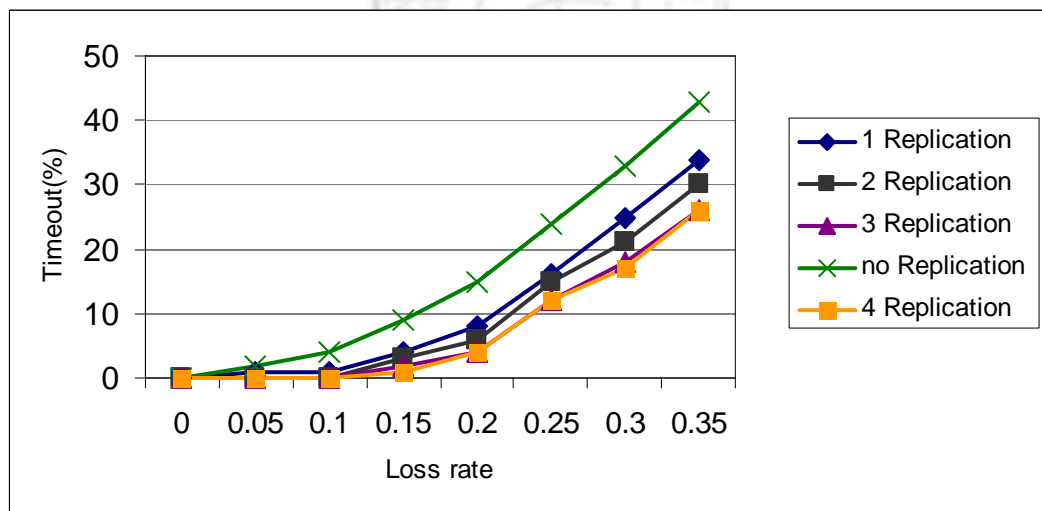


圖 3.5 No. of Replication V.S. Timeout Probability

圖 3.5 的 X 軸為封包遺失率，Y 軸為傳送端 Timeout 的次數，其中 1 Replication 為複製一個假封包，依此類推。從實驗中可以發現，在不同的封包遺失率環境中，隨著複製假封包數量的增加，傳送端 Timeout 的次數會減少。

當 Timeout 發生表示封包遺失必須重傳，而 Timeout 所造成的 overhead 可以

(3.1)表示:

$$\text{Overhead} = \text{No. of packets} \times \text{packet size} \times \text{loss rate} \times \text{retransmit times} \quad (3.1)$$

表 3.3 Replication overhead V.S. Timeout overhead

Replication	No. of packets	Replication overhead	Retransmission overhead	Total overhead
0	6,667	0KB	1,432KB	1,432KB
1	13,334	133KB	1,030KB	1,163KB
2	20,001	266KB	888KB	1,154KB
3	26,668	399KB	744KB	1,143KB
4	33,335	532KB	726KB	1,258KB

從表 3.3 可以看出，對於整體傳輸的 overhead 而言，retransmission overhead 佔了大部分，但是 total overhead 並不完全因為增加 Replication 而降低，四個 Replication 時的 total overhead 反而比三個 Replication 更高，所以在我們提出的 Replication Scheme 中我們設定 Replication 為三。

如果我們在真封包送出後隨即送假封包，有可能發生假封包比真封包先到達接收端的情形，當接收端先收到假封包的時候對於之後才收到的真封包會直接丟棄，這樣子可能會影響到整個服務的品質，所以我們將複製的假封包與真封包之間隔大於一個封包，讓真封包比假封包先到達的機會大幅提高。如圖 3.6 所示:

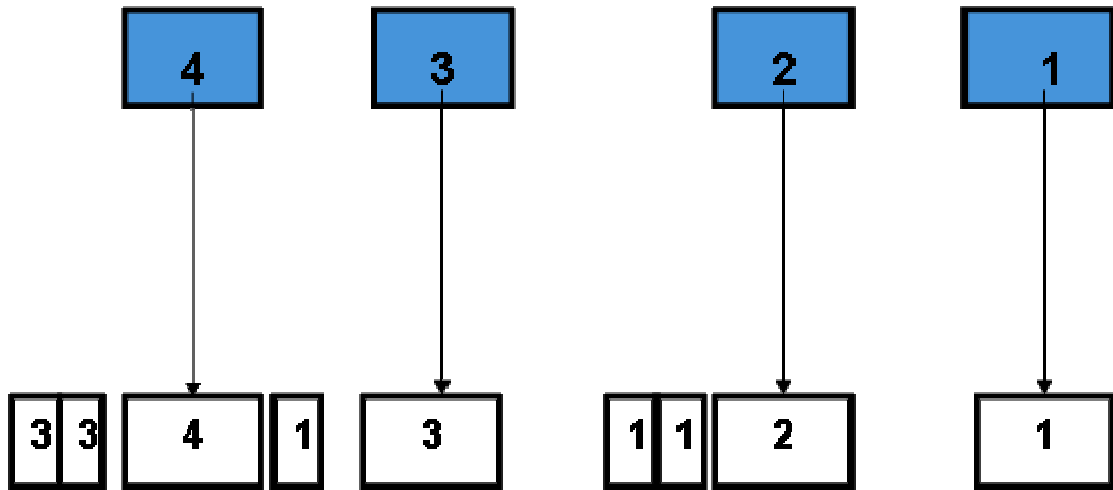


圖 3.6 Single-Side PR-TCP Replication Scheme

圖 3.6 舉例說明安插假封包的方式，其中封包 2 為 Registered class 的封包，用重傳機制保護，不使用假封包。封包 1、3、4 為 Regular class 的封包，在真封包之後傳送三個假封包，兩個安插於下一個真封包之後、一個安插於下兩個真封包之後。

藉由上法可以降低接收端因為收不到 ACK 而 Timeout 的機會，又不會因為接收端先收到假封包而丟棄真封包導致服務品質降低。表 3.4 為當封包遺失時，Single-Side PR-TCP 針對不同 Protection Class 所做的處理：

表 3.4 Packet Loss Handling Procedure in Single-Side PR-TCP

Protection Class	Sender	Receiver
Regular	retransmit duplicated packet	wait
Registered	retransmit	wait

因為我們無法控制接收端的 TCP，在重組封包時可能會發生錯誤(重組的封包中有假封包)，為了避免接收端的 TCP 發生錯誤，我們設定假封包 header 中 PSH 欄位值為 1。PSH 在 TCP header 的定義為:Push function，如果 PSH 為 1，此封包所攜帶的資料就會被直接遞給上層的應用程式毋需經過 TCP 處理。所以

上層的應用程式必須是 video streaming 這類可以不用確保所有資料傳送的應用服務。

3.4.3 Single-Side PR-TCP 的擁塞控制機制

Single-Side PR-TCP 的擁塞控制機制幾乎與 TCP 一樣，只有在 Fast Retransmit 時，Single-Side PR-TCP 重傳的封包不同，如上節所述。圖 3.7 為 Single-Side PR-TCP 的 state diagram、表 3.5 為每個事件觸發的狀態轉換以及接收端的 CWND 值變化情形：

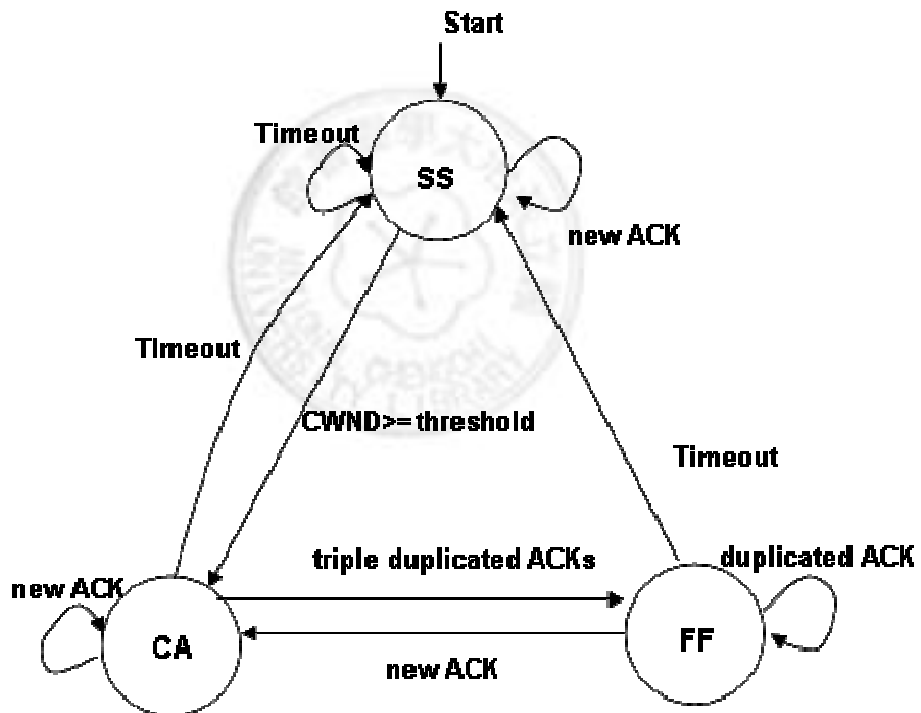


圖 3.7 State Diagram of Single-Side PR-TCP

表 3.5 State Transition in Single-Side PR-TCP

Event	Current State	Next State	Action
new ACK	(SS)	(SS)	$CWND = CWND * 2$
Timeout	(SS)	(SS)	$threshold = (1/2) * CWND$ $CWND = 1$
$CWND \geq threshold$	(SS)	(CA)	$CWND = CWND + 1$
new ACK	(CA)	(CA)	$CWND = CWND + 1$
Timeout	(CA)	(SS)	$CWND = 1$
triple duplicated ACKs	(CA)	(FF)	$threshold = CWND * (1/2)$ $CWND = threshold$
duplicated ACK	(FF)	(FF)	No change
new ACK	(FF)	(CA)	$CWND = CWND + 3$
Timeout	(FF)	(SS)	$CWND = 1$

藉由 Single-Side PR-TCP 的擁塞控制機制，可以有效的根據網路的狀況調節傳送速度，不致加重網路擁塞狀況。圖 3.8 為 Single-Side PR-TCP 接收端接收到 ACK 的流程，圖 3.9 為其演算法虛擬碼。

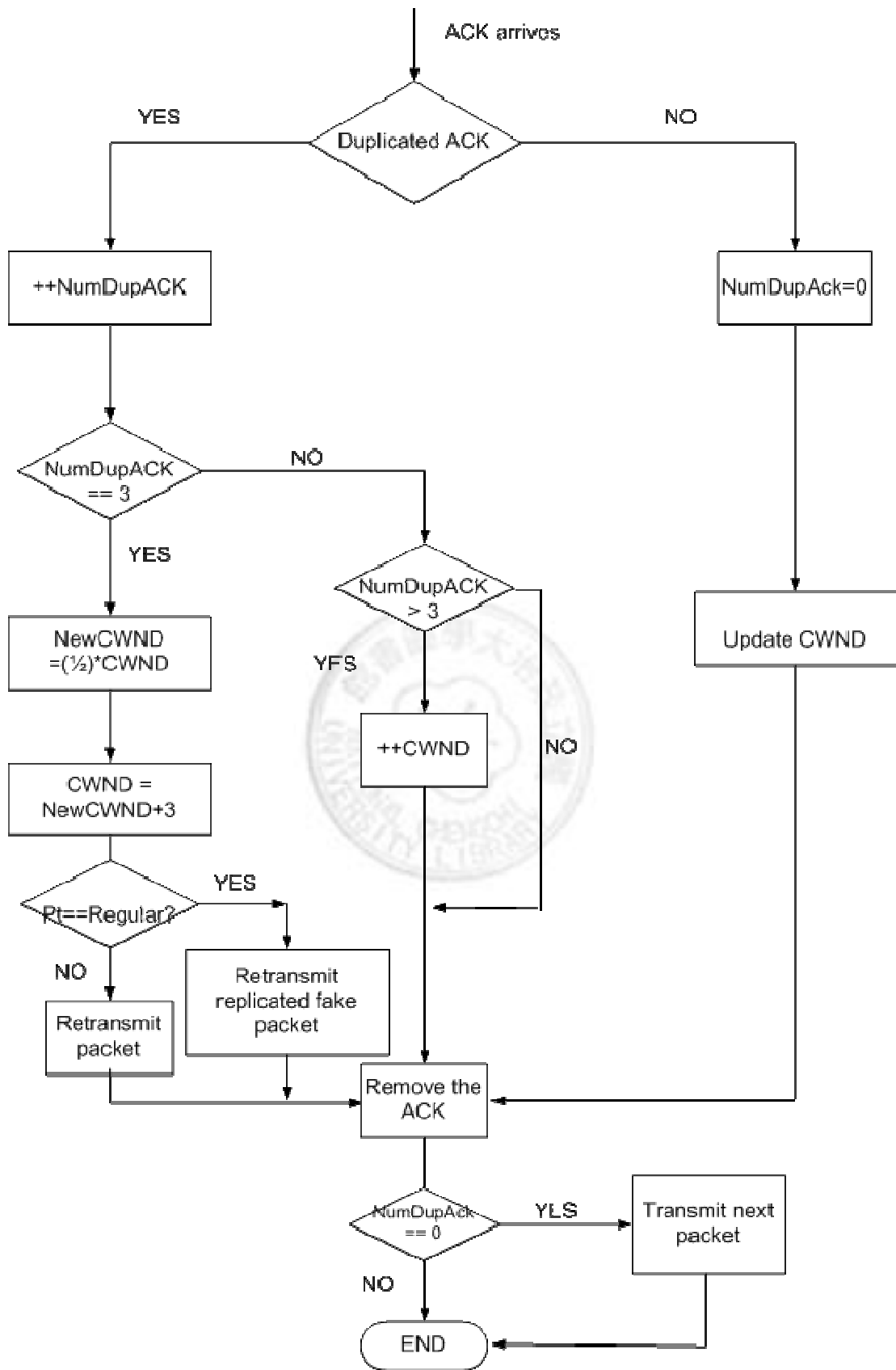


圖 3.8 ACK Arrival Procedure in Single-Side PR-TCP

```

ACKarrival(){
  If(duplicated ACK){
    NumDupAck++;
    If(NumDupAck==3 ){
      NewCWND=(1/2)*CWND;
      CWND=NewCWND+3;
      If(Pt==Regular){
        RetransmitreplicatedPacket( );
      }
      Else {
        RetransmitRealPacket( );
      }
    }
    Else if(NumDupAck>3){
      CWND++;
    }
    Else{
    }
  }
  Else{
    NumDupAck=0;
    UpdateCWND( );
  }
  RemoveACK( );
  If(NumDupAck==0)
  TransmitNextPacket( );
}

```

圖 3.9 Pseudo code for ACK Arrival Procedure in Single-Side PR-TCP

3.4.4 小結

Single-Side PR-TCP 只須放在傳送端便可以運行，特別是比起任何要改傳送端及接收端的通訊協定較容易推廣，適用於 client/server 的架構。為了降低接收端等待沒被重送的遺失封包的機會我們使用了 Replication Scheme，對於不保護的封包，複製 Header，直接重送，讓重傳的 overhead 減少並降低整體的平均傳送時間。

3.5 Packet Protection by Forward Error Correction

為了避免封包損傷導致太多的資料重傳，我們設計了一個自動重建遺失封包之機制(Packet Protection by Forward Error Correction)。在每 n 個封包之後，加上一個同位封包 (Parity Packet)，同位封包上的每一個 bit 是由 n 個封包中相對應的 bit 用同位計算的方式所產生。這 $n+1$ 個封包為一個錯誤更正的單位，稱為一個 Segment。而當一個 Segment 中任何一個封包發生遺失時(接收端已知 Segment 中遺失之封包編號)，就可利用同位封包將所遺失的封包還原。

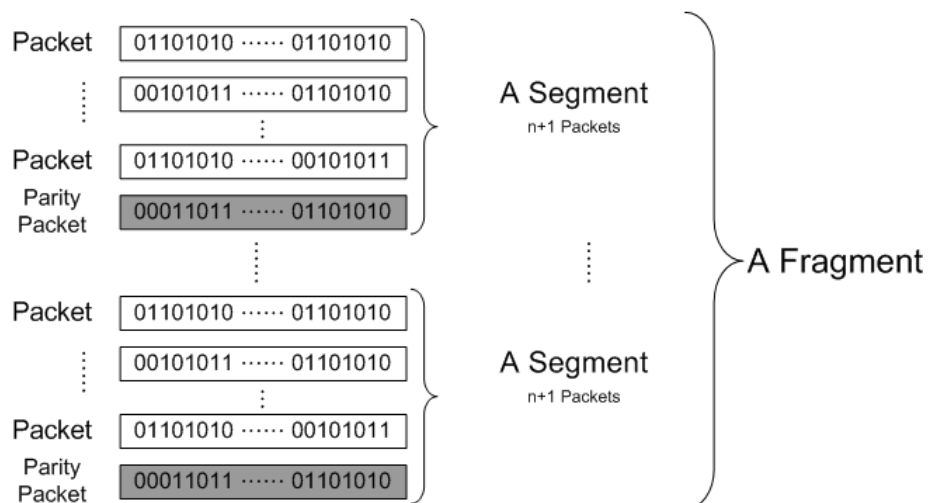


圖 3.10 Packet Protection by Forward Error Correction

若有超過一個封包出現錯誤，同位封包將無法彌補，如果是 Registered 或未過生命期的 Certified 封包則重傳，否則不重傳。

3.6 小結

在本章中，我們詳盡地介紹了我們提出的 PR-TCP，Basic PR-TCP 以及 Single-Side PR-TCP，對於整個 PR-TCP 的架構及運作機制也清楚的說明，我們將於下章利用實驗模擬，並與其它通訊協定做比較，來驗證我們的方法是可以有效的在網路資源較少的情況下提升 video streaming 這類應用服務的品質。



第四章

效能評估

我們將於本章以即時影音傳輸為例將我們的 PR-TCP 與現行的通訊協定(如 TCP Reno、UDP、TFRC 等)做比較。

4.1 實驗環境

我們以 NS-2 (Network Simulator - version 2)[31,32]網路模擬器作為實驗模擬環境，並觀察 video 封包在各種通訊協定控制下的傳輸結果，使用 Evalvid[11,27]這套多媒體影像傳輸系統來做模擬，實際觀察 video 品質。

Evalvid 是一套多媒體品質評估工具，提供了使用者從原始影像資料壓縮、封包傳送產生效能分析結果，在視覺上比較影像傳輸品質的好壞。圖 4.1 為 Evalvid 架構:

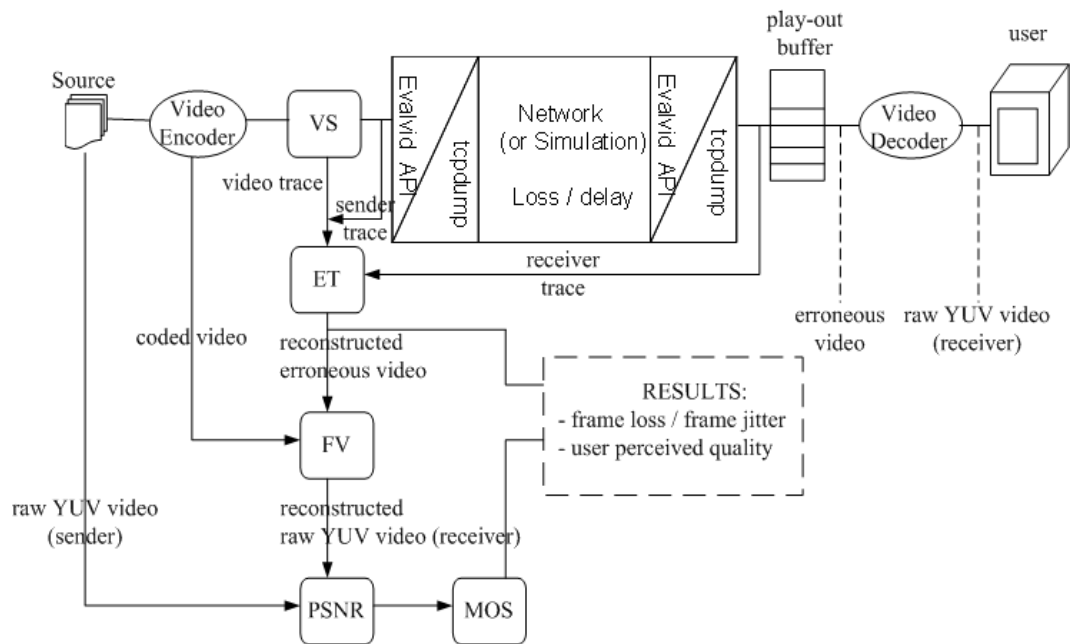


圖 4.1 Structure of Evalvid

1. 將 Raw YUV[33] video 編碼成 MPEG-4 的格式，並轉換成 NS2 外部流量的 traffic trace file、產生封包 ID、傳送時間、封包大小的 script 檔供 NS2 分析。
2. 經由設定的網路環境進行模擬後產生兩個檔案 sender 與 receiver trace files，sender 記錄傳送時的封包 ID、順序及發送時間，receiver 紀錄接收的封包 ID、順序及接收時間。
3. 透過產生出來的 trace file 可以得知所有的封包傳輸資訊，藉由這些資訊可以計算傳送時間。
4. 我們將根據 trace file 將網路模擬的結果，重建新的 MPEG-4 格式的檔案，再解碼還原成 YUV 的 video。
5. 透過 PSNR 值計算以及影像觀察比較經由網路模擬後以及原來 Raw YUV video 的畫質差異。

4.2 實驗評估指標：

Video streaming 的基本概念是:邊下載邊播放「Play as received」,不同的應用, delay bound(最大容忍封包延遲時間)的限制也不同,例如 video conference 應用的 delay bound 為 600ms, VoD 或是其他即時性要求不那麼嚴苛的應用, delay bound 就比較大。

影響 video streaming 品質的關鍵在於能在其 delay bound 內送達接收端的可解壓縮畫面數,不管送的快送的慢,或是重傳幾次,只要在 delay bound 內送達,都能符合應用程式的需求。例如:畫面 i 的送出時間是第 0.1 秒,則接收端要在 0.7 秒之前收到該畫面,才能確保影像的品質。以下為我們的評估指標:

量化指標:

A. 可解畫面封包數 (Number of Packets in Decodable Frames):

一個畫面在 delay bound 內到達接收端的封包數量必須超過一個門檻,才可被解壓縮,此外,這個畫面所參考的畫面也要是可被解壓縮的。畫面可解壓縮的條件如下所示:

- I-frame:只要收到的封包數量超過門檻即可解壓縮。
- P-frame:封包數量超過門檻,而且所參考的 I-frame 必須可解壓縮。
- B-frame:封包數量超過門檻,而且所參考的 I-frame, P-frame 必須可解壓縮。

可解畫面封包數必須符合下述條件:

1. (封包的到達時間-封包的送出時間) < delay bound
2. 必須是有效封包,且參考的封包也必須是有效封包

我們以可解畫面封包數當做我們的參考指標,可解畫面封包數越高,影像品質越好。

B. PSNR (Peak Signal to Noise Ratio):

PSNR 是一個常用來評比畫質的比較工具，通常以對數分貝(dB, decibel)單位來表示。

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (4.1)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2 \quad (4.2)$$

PSNR 主要透過均方差 (Mean Square Error, MSE)計算出來。假設兩個 $m*n$ 的單色圖像 I 與 K 的噪聲(Noise)相近，他們的均方差定義為公式(4.2)，PSNR 則定義為公式(4.1)。其中 MAX_I 表示圖像點最大值，若每個採樣點用 8 bit 表示，那麼就是 255。圖像壓縮中典型的 PSNR 在 30 到 40dB 之間。表 4.1 將 PSNR 值對應到 5 個 MOS (Mean Opinion Score)級別來進行質量評價，我們可以很直觀地瞭解 PSNR 值的品質為何。

表 4.1 PSNR and MOS

PSNR[dB]	MOS
>37	5 (Excellent)
31-37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
<20	1 (Bad)

C. 額外耗用的網路資源 (Wasted Transmission Overhead)

對 video streaming 這類應用服務來說，過了 delay bound 送達接收端的封包是無效的，傳輸這些逾時的封包，會浪費網路資源。

無效封包(Invalid Packet)的定義:送達接收端，但是超出 delay bound 的封包。

額外耗用的網路資源計算方式如下:

$$WTO = (\text{no. of Invalid packets} + \text{no. of retransmitted packets}) \times \text{packet size} \quad (4.3)$$

4.3 實驗設計

實驗的拓樸是由 2~10 個節點組成的有線/無線網路環境，模擬影音的傳輸，以 Foreman 及 Container 為實驗的輸入影像，並在實驗中加入 burst 性質的訊務以增加網路負載的變化，我們依各個子實驗調整不同的參數(表 4.2)並觀察評估指標，以 TCP Reno、UDP 及 TFRC 做為對照組，比較其優劣，並比較 PR-TCP 的兩個版本:Basic version、Single-Side version 及不同的保護方案(僅保護 I frames，保護 I、P frames)之影像品質差異。



圖 4.2 Simulation Video:Foreman 、 Container

表 4.2 Parameters of Simulation A and B

Parameter	Range
No. of hops	2~10
Delay bound	0.6 sec~8 sec
Queue size	20 packets
Loss rate	0~0.5
Burst traffic load	500Kbps
Certified class	I-frame , I/P frame

4.4 實驗 A:有線網路

4.4.1 實驗目標

本實驗的目標是為了驗證我們的方法可以在條件受限且變動的有線網路環境中，在量化的評估指標上可以較其他的通訊協定有較好的表現。

4.4.2 實驗流程



圖 4.3 Topology of Simulation A

表 4.3 Parameters of Simulation A

Parameter	Range
No. of hops	2~10
Delay bound	0.6 sec~8 sec
Queue size	20 packets
Certified class	I-frame, I&P frame

圖 4.3 為我們實驗 A 的拓樸，表 4.3 為實驗 A 的參數表，我們建立了 source、destination 及數個路由器(依照實驗不同增加不同數量的路由器，2~10 個)組成的拓樸圖，使用 Foreman 及 Container 作為傳輸的標的影像，並且第 4 秒及第 9 秒在 router A 及 router B 之間有 burst traffic (500kbps)干擾，使用的通訊協定則依我們要實驗的目標而改變(例如 TCP Reno、UDP、...)，並調整 delay bound，模擬時間為 15 秒，router 和 router 之間的頻寬是 5Mb，延遲時間 1ms，source 到 router A 及 router B 到 destination 的頻寬是 10Mb，延遲時間 1ms，路由器佇列的管理機制為 DropTail，queue size 為 20 packets。主要觀察平均延遲時間、可解畫面封包數及 PSNR 值。實驗 A 要比較的對象為：

- PR(I): Basic PR-TCP: 僅將 I-frame 設成 Certified class，其餘 P-frame、B-frame 皆為 Regular class。

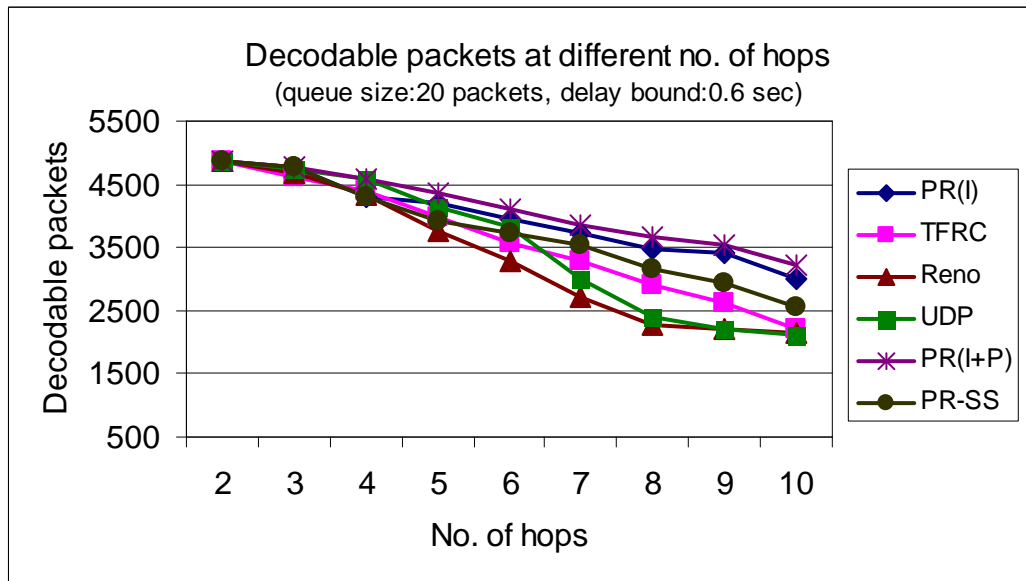
- PR(I+P): Basic PR-TCP:將 I-frame 及 P-frame 設成 Certified class，B-frame 定為 Regular class。
- PR-SS: Single-Side PR-TCP:將 I-frame 設成 Registered class，其餘 P-frame、B-frame 皆為 Regular class。
- TCP Reno
- UDP
- TFRC

4.4.3 實驗結果分析

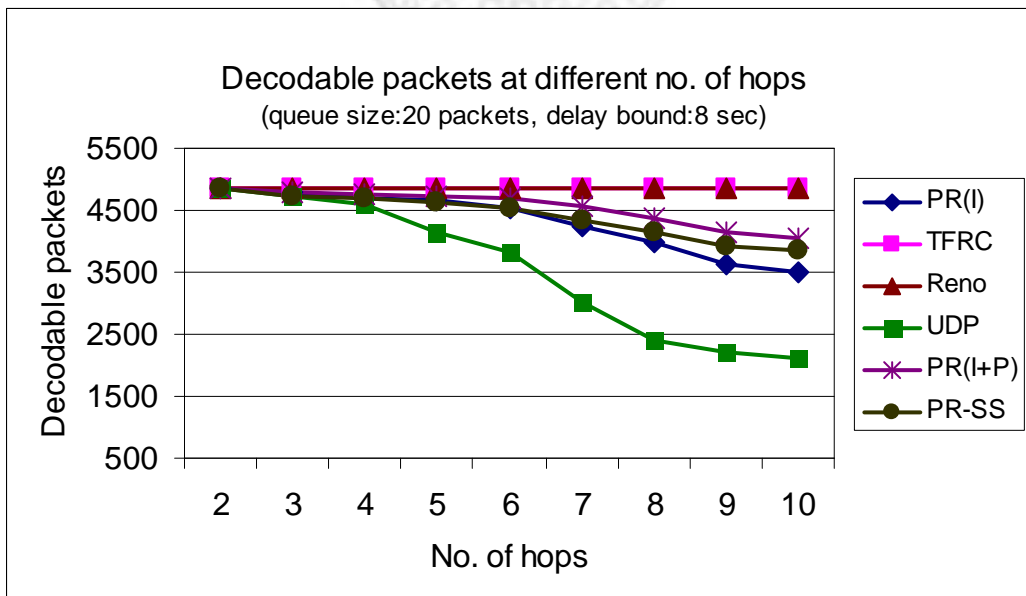
使用 Foreman 及 Container 作為傳輸標的影像的實驗結果差異不大，論文中只展示 Foreman 的實驗結果。從圖 4.4、4.5 及 4.6 可以觀察出在 delay bound 較小(如 video conference 應用)、hop 數在 6~10(網路情況較差)的時候，PR(I+P)比其他方案的可解畫面封包數較多、PSNR 值較高、頻寬浪費較少。因為只保護重要性較高的封包(如 I、P-frame)，不重傳不重要的遺失封包(如 B-frame)，封包能在 delay bound 內送達的比較多。UDP 因為不重傳，當重要封包遺失時，會影響到其他封包解碼，對影像品質造成極大的影響。TCP Reno 及 TFRC 會重傳每個遺失封包，可能導致很多封包逾期送達，變成無效封包，而降低影像品質，而重傳這些無效封包會造成更嚴重的頻寬浪費。PR-SS 會多耗用一點點的網路資源傳送假封包，但是能有效減少重傳次數，降低重傳的 overhead。

相同的網路狀況下，當 delay bound 較大的時候(如 VoD 應用)，PR(I+P)的可解畫面封包數及 PSNR 值可能會比 TCP Reno、TFRC 較低，因為 TCP Reno 及 TFRC 有足夠的時間確保所有封包到達，若沒有 delay bound 的限制，則 TCP Reno 及 TFRC 的影像品質是最好的，但是 PR(I+P)比 TCP Reno、TFRC 重傳次數少，額外耗用的頻寬較少。

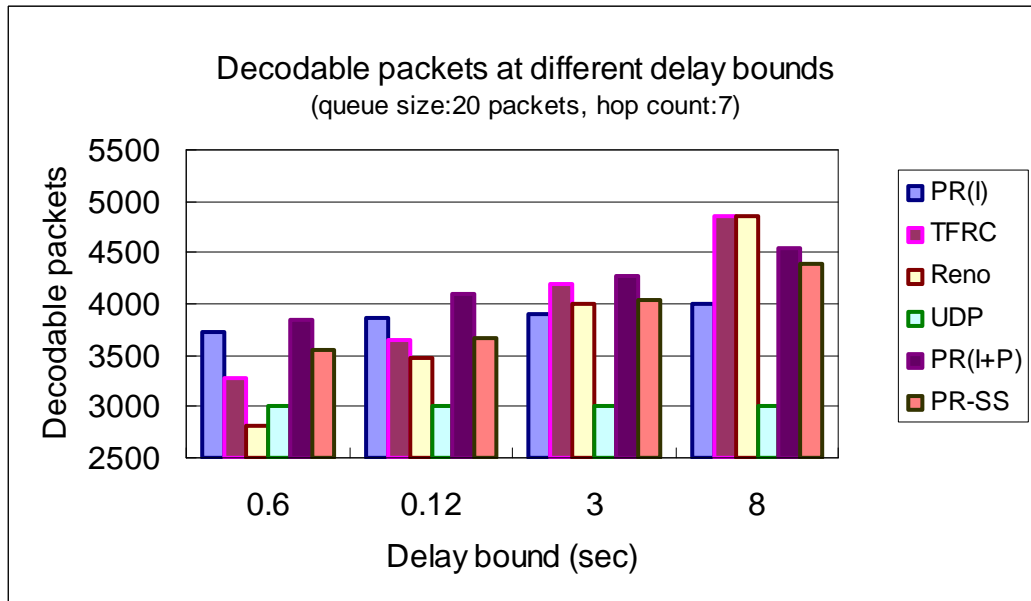
在網路情況較差的環境中，Basic PR-TCP 比起其他通訊協定在傳輸 delay bound 較低的即時性影音串流服務有較好的表現，而在傳輸 delay bound 較大的影音串流服務時雖然影像品質稍差於 TCP Reno 及 TFRC，但是節省了較多的頻寬資源。



(a)



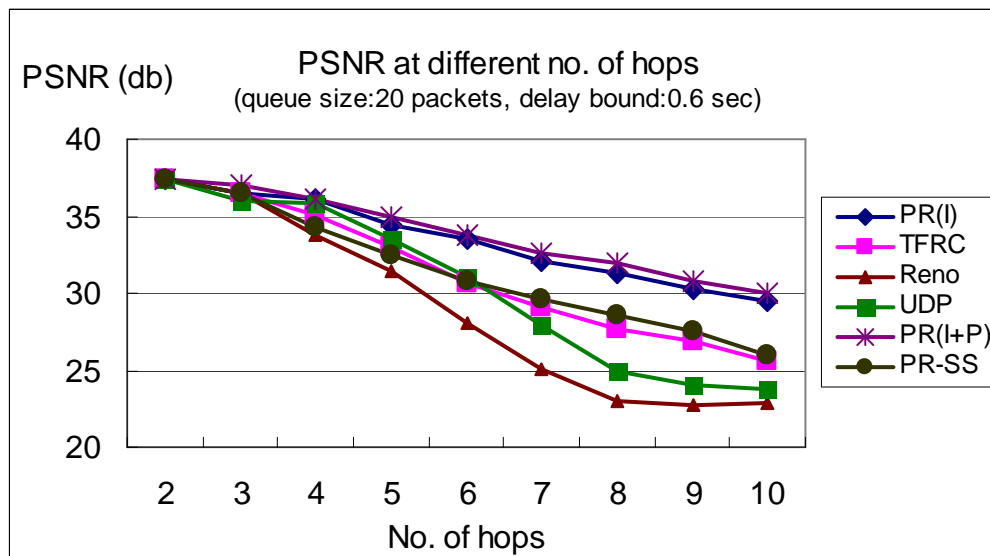
(b)



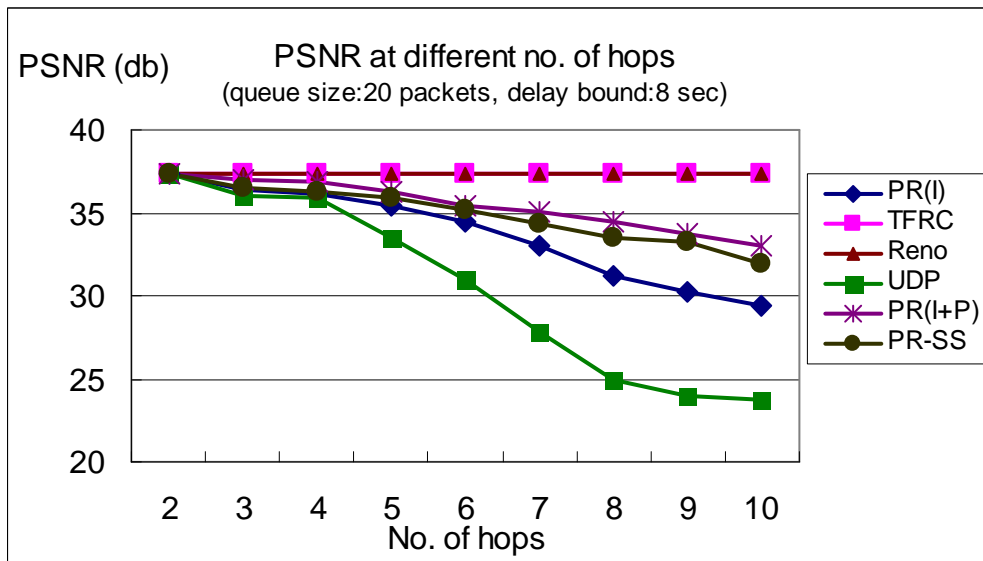
(c)

圖 4.4 實驗 A 結果: 各通訊協定在不同 hop 數中 decodable packets 的比較

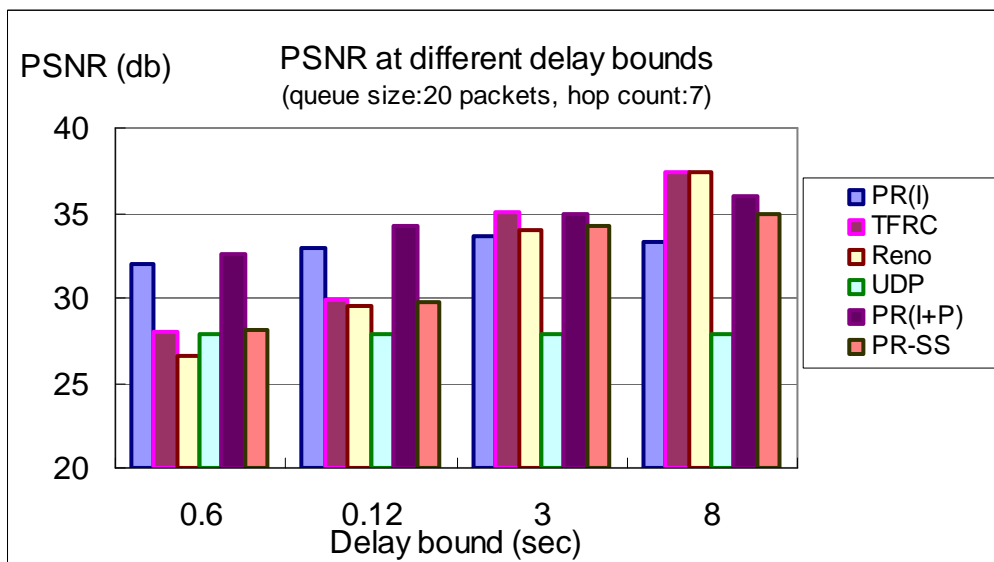
從圖 4.4(a)可以觀察到當 delay bound 設為 0.6 sec, hop 數在 6~10 時, PR(I+P) 比其它方案的可解畫面封包數最少多 18%, PR-SS 的可解畫面封包數比 PR(I+P) 少了約 10%。從圖 4.4(b)、(c)可以觀察到當 delay bound 設為 8 sec、hop 數在 6~10 時, TCP Reno 及 TFRC 的可解畫面封包數最高, 因為兩者皆確保所有封包到達。PR(I+P)因為只重傳保護的遺失封包, 雖然可解畫面封包數增加, 但是增加的幅度不如 TCP Reno 及 TFRC。



(a)



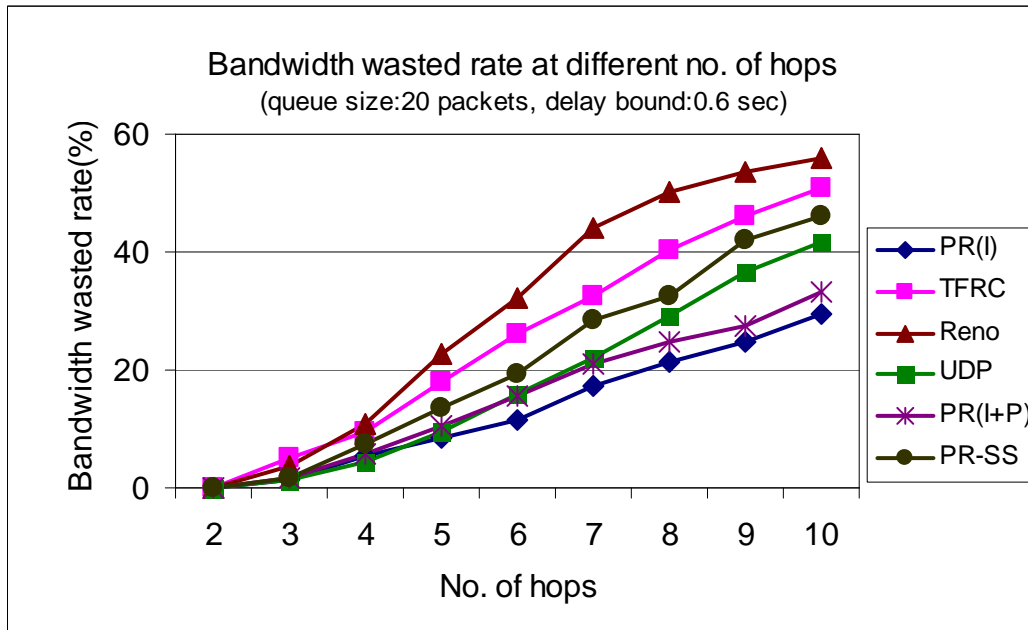
(b)



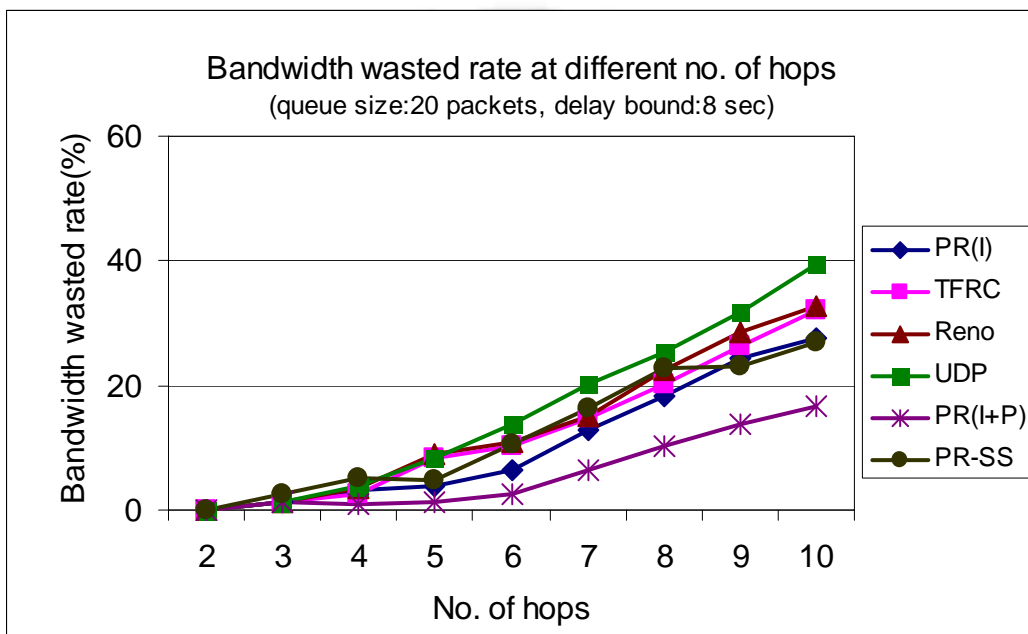
(c)

圖 4.5 實驗 A 結果: 各通訊協定在不同 hop 數中 PSNR 值的比較

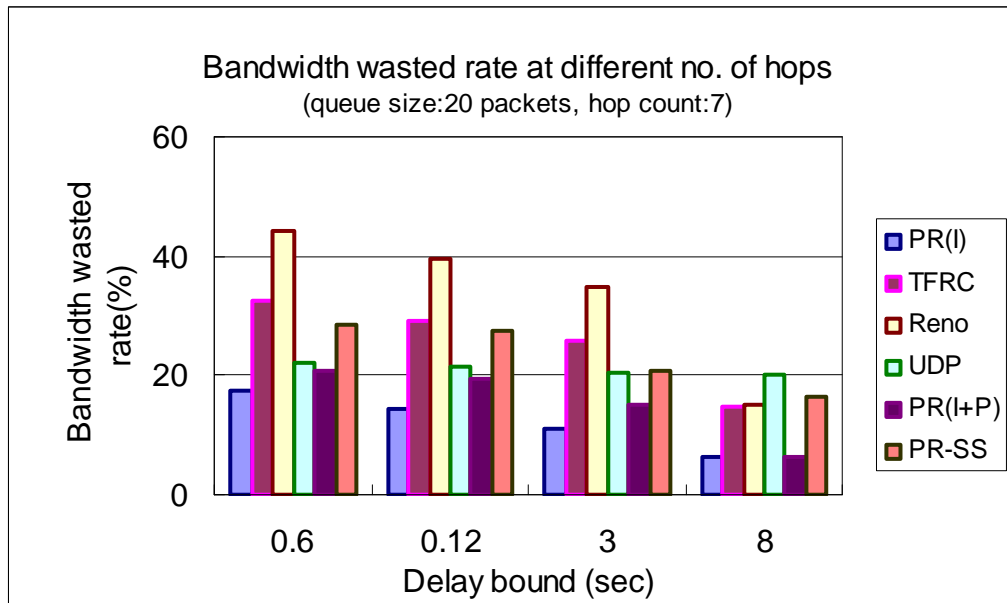
從圖 4.5(a)可以觀察到當 delay bound 設為 0.6 sec, hop 數在 6~10 時, PR(I+P) 方案比其它方案的 PSNR 值最少多 15%, PR-SS 的 PSNR 值比 PR(I+P) 少了約 11%。從圖 4.5(b)、(c)可以觀察到當 delay bound 設為 8 sec, hop 數在 6~10 時, TCP Reno 及 TFRC 的 PSNR 值最高。PR(I+P) 的 PSNR 值增加的幅度不如 TCP Reno 及 TFRC。



(a)



(b)



(c)

圖 4.6 實驗 A 結果: 各通訊協定在不同 hop 數中額外耗用的網路資源比較

我們將額外耗用的網路資源 normalize(以模擬的影像資料大小為分母)。

從圖 4.6(a)可以觀察到當 delay bound 設為 0.6 sec, hop 數在 6~10 時, PR(I+P) 比 TCP Reno 及 TFRC 最少節省 12% 的頻寬資源, PR-SS 比 PR(I+P) 多耗用了約 10% 的頻寬資源。從圖 4.6(b)、(c)可以觀察到當 delay bound 設為 8 sec、hop 數在 6~10 時, PR(I+P) 比 PR-SS、TCP Reno 及 TFRC 最少節省 8% 的頻寬資源, 此時 TCP Reno 及 TFRC 每個封包都是可解畫面封包, 對於應用程式而言只有重傳的 overhead, 所以額外耗用的網路資源減少了。

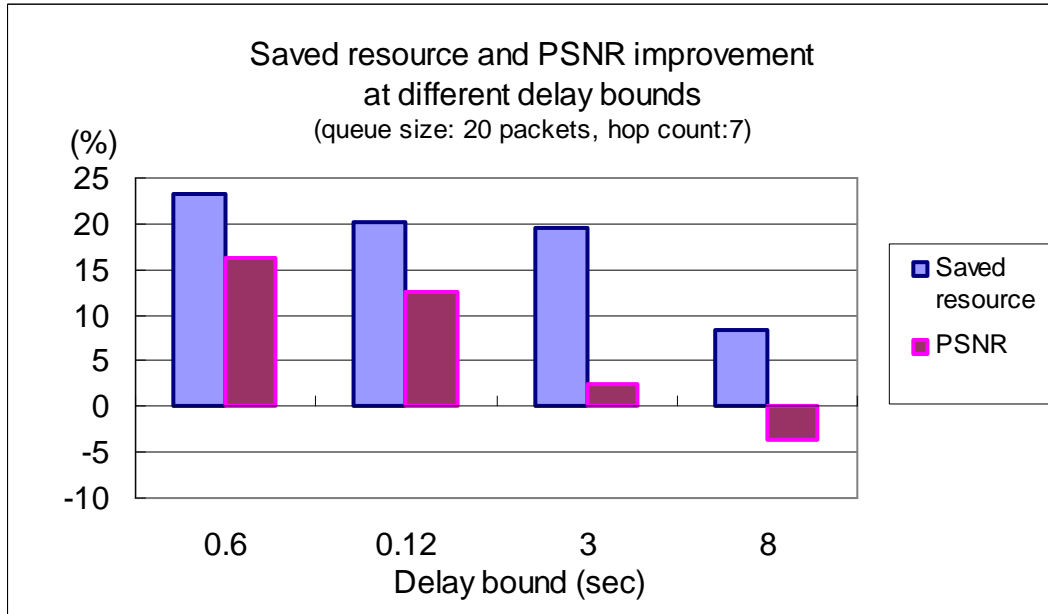


圖 4.7 實驗 A 結果: PR-TCP 比 TCP Reno 在 Saved resource 及 PSNR 提升比較

從圖 4.7 可以看出在 delay bound 較小的狀況，PR-TCP 比 TCP Reno 節省約 24% 的網路資源，而 PSNR 值約提升 16%，而在 delay bound 較大的情況下，雖然 PR-TCP 比 TCP Reno 減少了約 3% 的 PSNR 值，但是節省了約 9% 的網路資源。

4.5 實驗 B: 有線網路+無線網路

4.5.1 實驗目標

本實驗的目標是為了驗證我們的方法在條件受限且變動的有線及無線的網路環境中，仍能維持較好的表現。

4.5.2 實驗流程

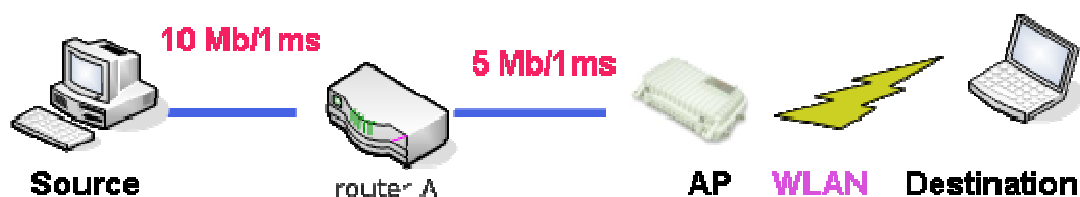


圖 4.8 Topology of Simulation B

表 4.4 Parameters of Simulation B

Parameter	Range
No. of hops	4
Delay bound	0.6 sec~8 sec
Queue size	20 packets
Loss rate	0~0.5
Certified class	I-frame , I&P frame

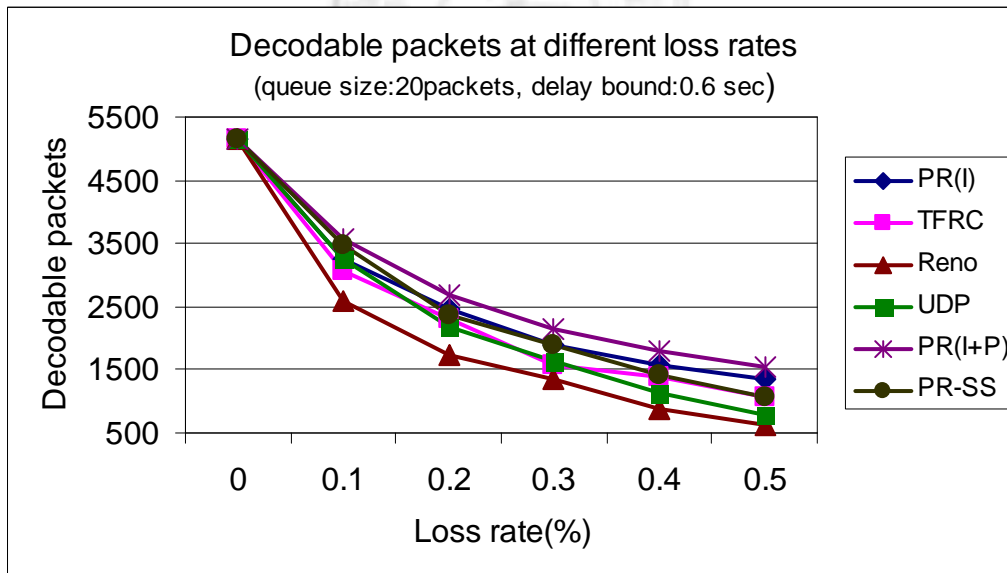
圖 4.8 為實驗 B 的拓樸，表 4.4 為實驗 B 的參數，我們建立有線及無線網路的網路環境，使用 Foreman 及 Container 為輸入影像，在第 4 秒及第 9 秒在 router A 及 AP 之間有 burst traffic (500kbps)作為干擾，使用的通訊協定則依我們要實驗的目標而改變(例如 TCP Reno、UDP、...)，並調整 delay bound，模擬時間為 15 秒，source 到 router A 的頻寬是 10Mb，延遲時間 1ms，router A 和 AP 之間的頻寬是 5Mb，延遲時間 1ms，路由器佇列的管理機制為 DropTail，queue size 為 20 packets。跟實驗 A 不同的地方在於，我們控制 AP 到 destination 之間的封包遺失率(0~0.5)。要觀察的指標以及對像則跟實驗 A 相同。

4.5.3 實驗結果分析

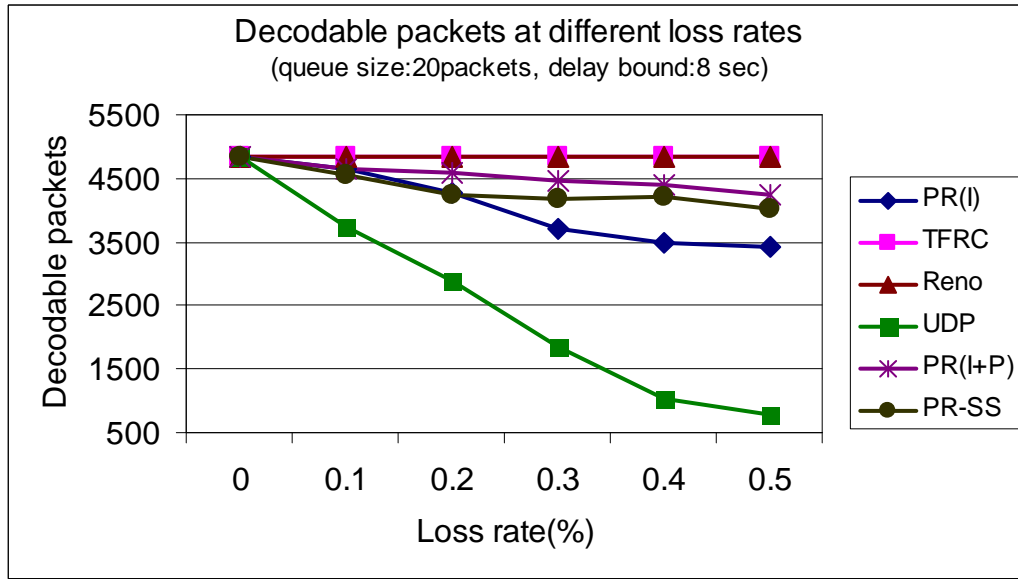
Foreman 及 Container 的結果差不多，我們只展示 Foreman 的實驗結果。從圖 4.9、4.10 及 4.11 可以觀察出在 delay bound 較小、loss rate 在 0.3~0.5(封包遺失率較高)的時候，PR(I+P)及 PR(I)比其他方案一樣能有更好的表現，但是當封包遺失太嚴重的時候，改善的程度就沒有那麼大。

相同的網路狀況下，當 delay bound 較大的時候，PR(I+P)的可解畫面封包數及 PSNR 值可能會比 TCP Reno、TFRC 較低，但是 PR(I+P)能比 TCP Reno、TFRC 節省更多頻寬資源。

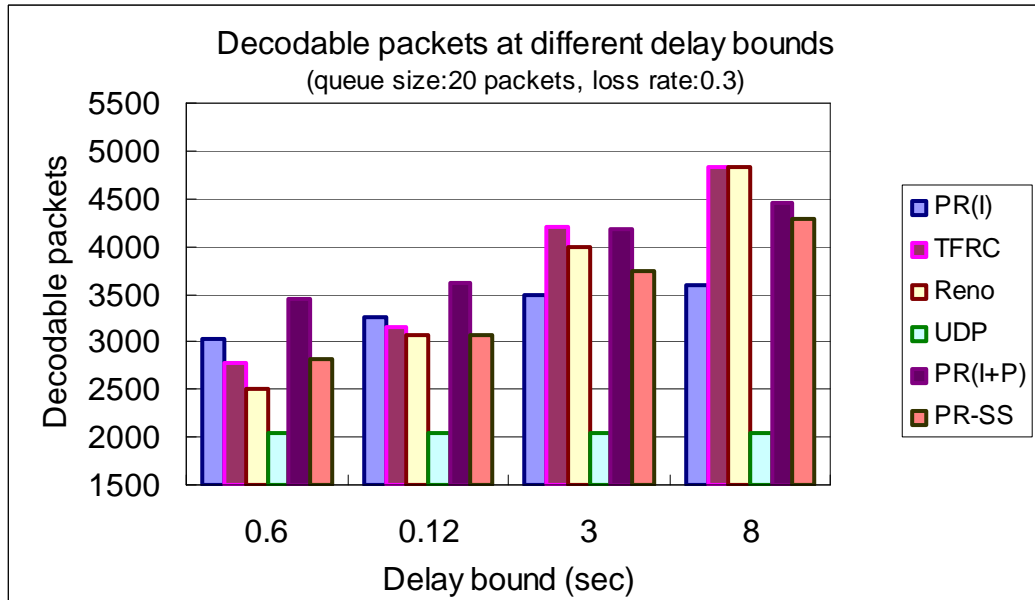
在無線的網路環境中，Basic PR-TCP 比起其他通訊協定在傳輸 delay bound 較低的即時性影音串流服務仍能有較好的表現，在傳輸 delay bound 較大的影音串流服務時雖然影像品質差於 TCP Reno 及 TFRC，但是節省了更多的頻寬資源。



(a)



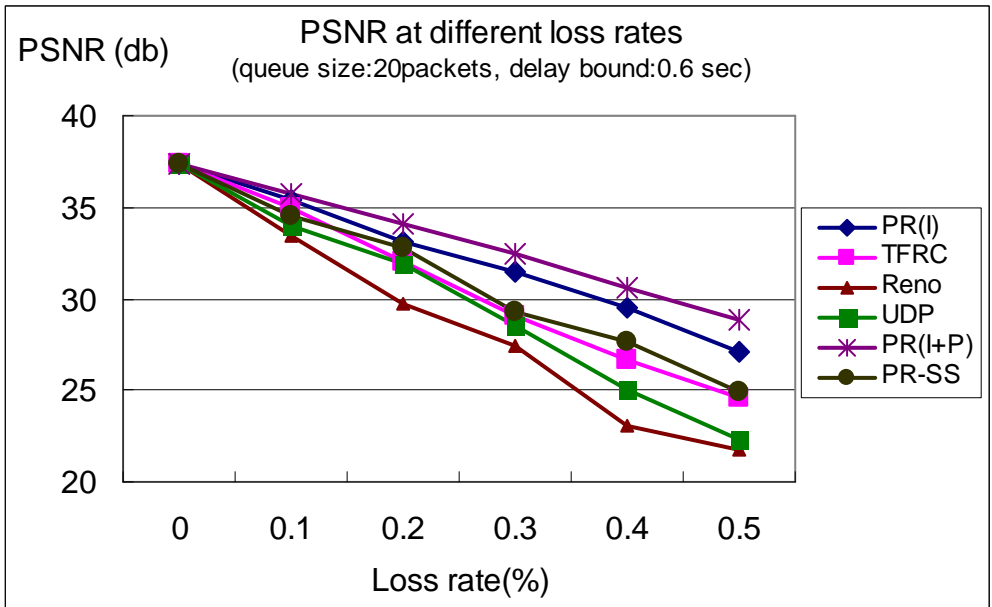
(b)



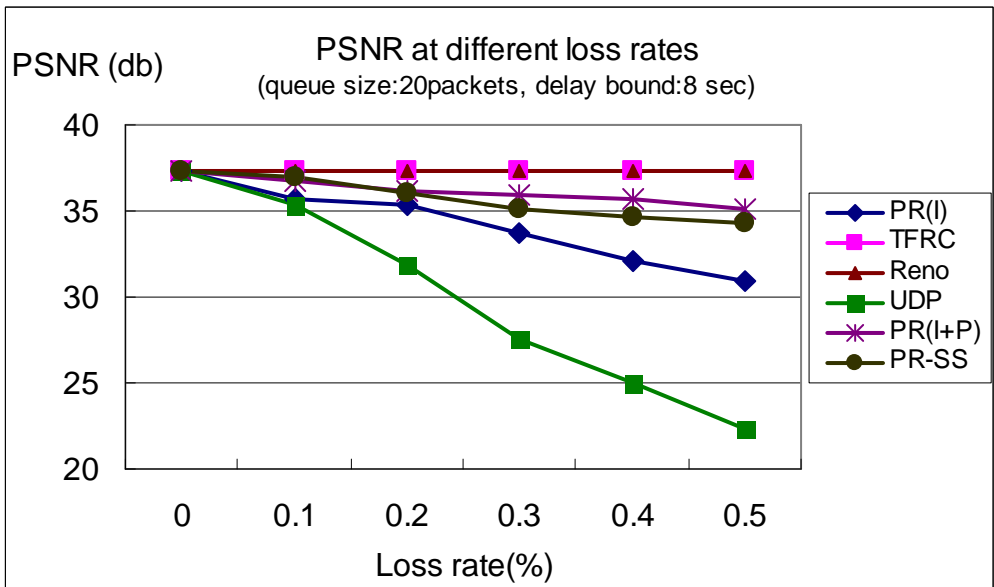
(c)

圖 4.9 實驗 B 結果: 各通訊協定在不同 loss rate 中 decodable packets 的比較

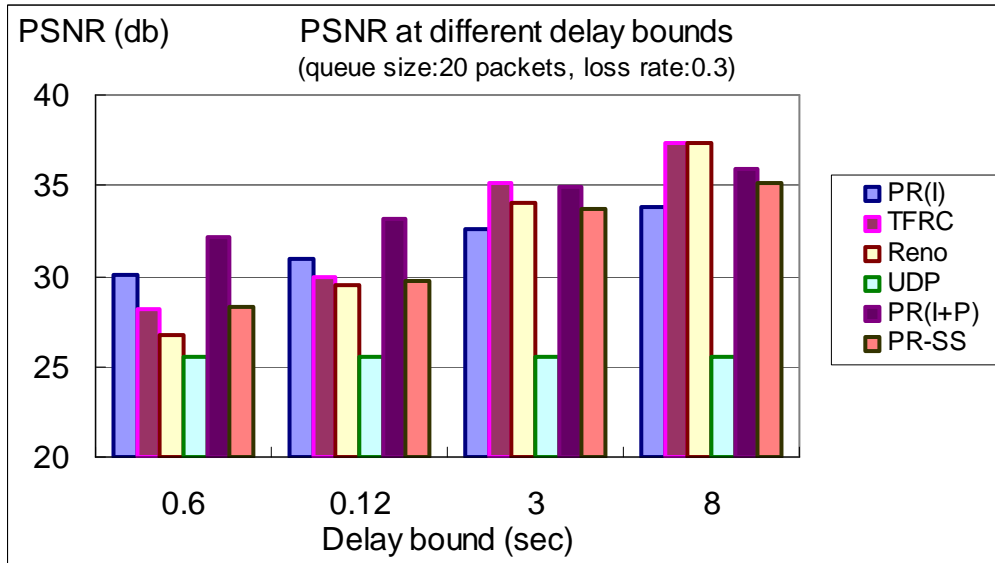
從圖 4.9(a)可以觀察到當 delay bound 設為 0.6 sec，loss rate 在 0.3~0.5 時，PR(I+P)比其它方案的可解畫面封包數最少多 19%，PR-SS 的可解畫面封包數比 PR(I+P)少了約 15%。從圖 4.9(b)、(c)可以觀察到當 delay bound 設為 8 sec，loss rate 在 0.3~0.5 時，TCP Reno 及 TFRC 的可解面封包數最高，PR(I+P)的可解畫面封包數略低於 TCP Reno 及 TFRC。



(a)



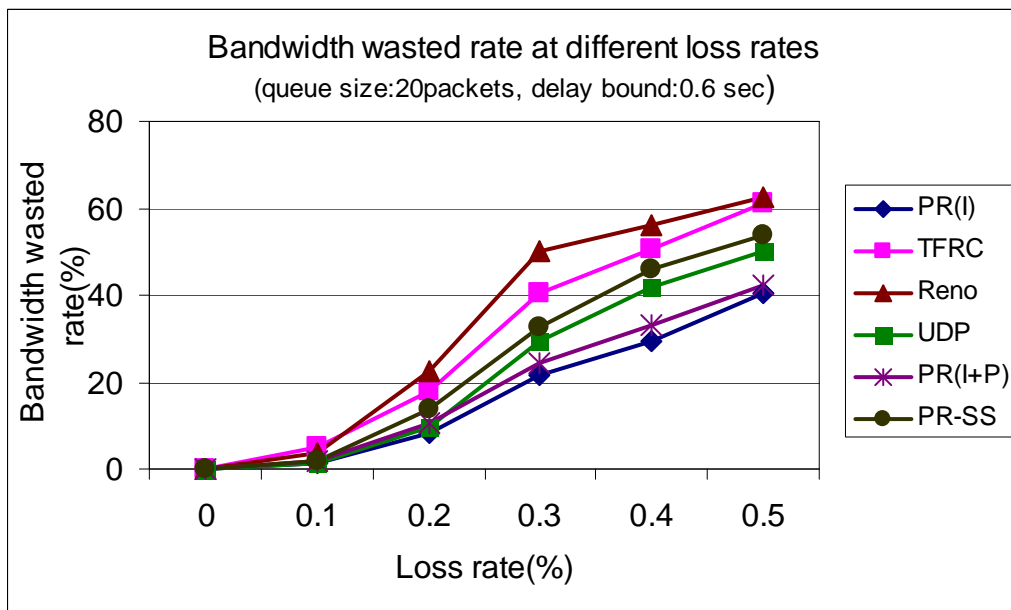
(b)



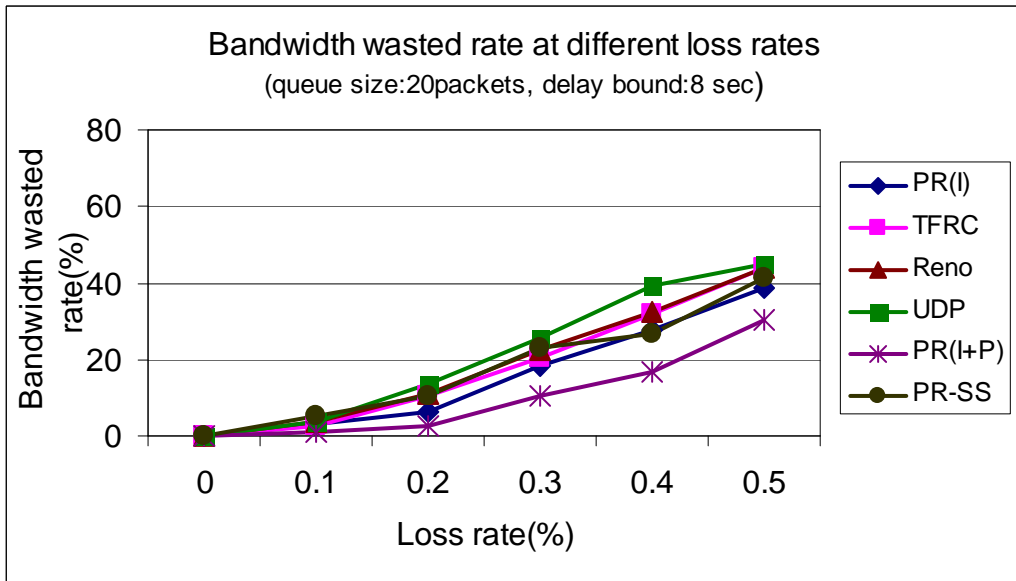
(c)

圖 4.10 實驗 B 結果: 各通訊協定在不同 loss rate 中 PSNR 值的比較

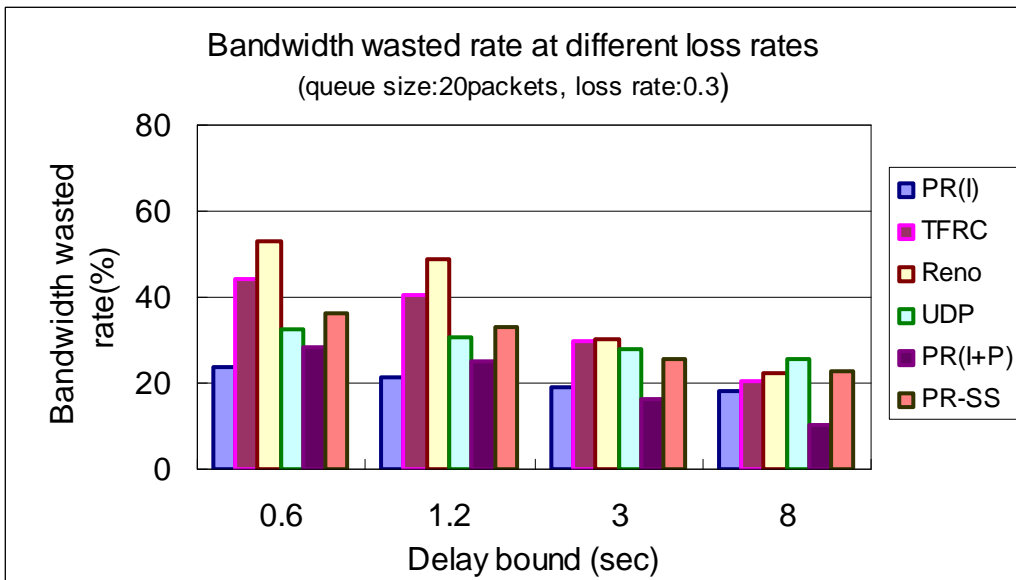
從圖 4.10(a)可以觀察到當 delay bound 設為 0.6 sec，loss rate 在 0.3~0.5 時，PR(I+P)比其它方案的 PSNR 值最少多 20%，PR-SS 的 PSNR 值比 PR(I+P)少了約 14%。從圖 4.10(b)、(c)可以觀察到當 delay bound 設為 8 sec，loss rate 在 0.3~0.5 時，TCP Reno 及 TFRC 的 PSNR 值最高，PR(I+P)的 PSNR 值略低於 TCP Reno 及 TFRC。



(a)



(b)



(c)

圖 4.11 實驗 B 結果: 各通訊協定在不同 loss rate 中 bandwidth wasted rate 的比較

我們將額外耗用的網路資源 normalize(以模擬的影像資料大小為分母)。

從圖 4.11(a)可以觀察到當 delay bound 設為 0.6 sec, loss rate 在 0.3~0.5 時, PR(I+P)比 PR-SS、TCP Reno 及 TFRC 最少節省 15%的頻寬資源。從圖 4.11(b)、(c)可以觀察到當 Delay bound 設為 8 sec、loss rate 在 0.3~0.5 時, PR(I+P)也比 PR-SS、TCP Reno 及 TFRC 最少節省 10%的頻寬資源。

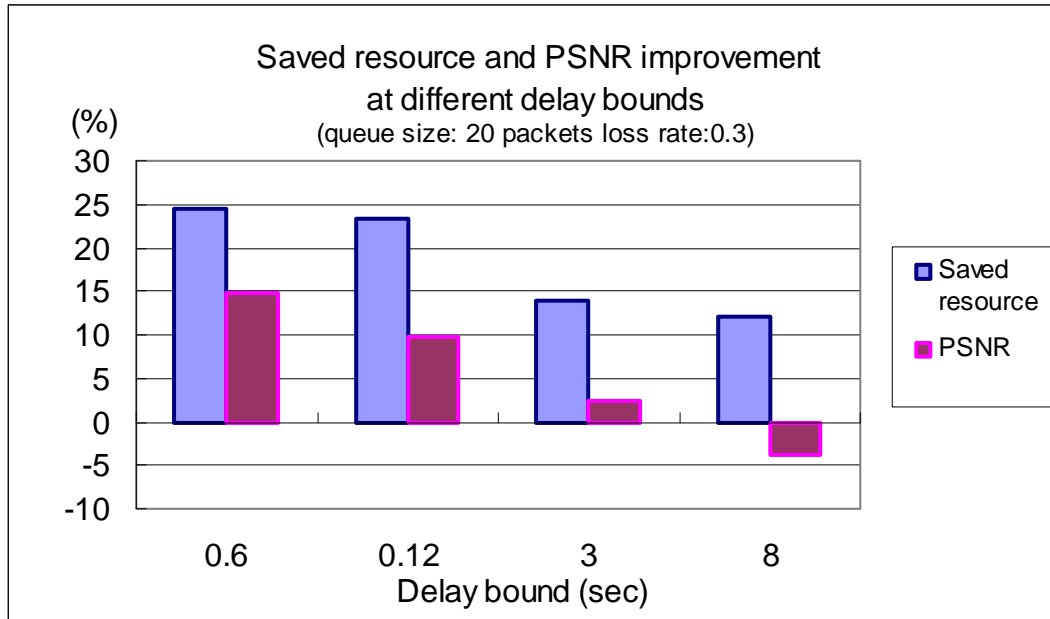


圖 4.12 實驗 B 結果: PR-TCP 比 TCP Reno 在 Saved resource 及 PSNR 提升比較

從圖 4.12 可以看出在 delay bound 較小的狀況，PR-TCP 比 TCP Reno 節省約 25%的頻寬，而 PSNR 值約提升 15%，而在 delay bound 較大的情況下，雖然 PR-TCP 比 TCP Reno 減少了約 4%的 PSNR 值，但是節省了約 12%的頻寬。

4.6 實驗 C:影像品質評估

4.6.1 實驗目標

我們觀察經由網路模擬後的影像觀察來比較 PR-TCP 及其他通訊協定。

4.6.2 實驗流程

我們以實驗 A 的拓樸作為實驗 C 的拓樸(以 Foreman 為輸入影像)，在 hops 數為 5 的情況下，擷取第 6 秒當 burst traffic 正在干擾網路時的影像畫面做分析比較。

4.6.3 實驗結果分析

從圖 4.13~圖 4.18 可以看出畫面品質的差異程度，圖 4.13 為原始的畫面，圖 4.14 為使用 TCP Reno 傳輸所得畫面，人臉幾乎無法辨識。圖 4.15、4.16 分別為使用 UDP 及 Single-Side PR-TCP 傳輸所得畫面，兩張畫面差別不會太大，同是在帽緣，臉頰及嘴巴的部份不清楚。圖 4.17 為使用 TFRC 傳輸所得畫面，只有在帽緣，臉頰比較模糊，且比 TCP Reno、UDP、Single-Side PR-TCP 畫面較為平順。圖 4.18 為使用 Basic PR-TCP(I+P)傳輸所得畫面，可以觀察到，只有某些區塊有點模糊，但是整張畫面辨識度是非常高的。



圖 4.13 Original frame #111



圖 4.14 實驗 C 結果: Received Frame #111 by using TCP Reno



圖 4.15 實驗 C 結果: Received Frame #111 by using UDP



圖 4.16 實驗 C 結果: Received Frame #111 by using Single-Side PR-TCP



圖 4.17 實驗 C 結果: Received Frame #111 by using TFRC



圖 4.18 實驗 C 結果: Received Frame #111 by using Basic PR-TCP



第五章

結論

某些網路服務的資訊封包具有不同的重要性而要求不同程度的保護，但現行的通訊協定，如 TCP、UDP 並不提供彈性的封包保護，前者提供完全保護而後者完全不提供保護，不適用於上述的網路服務。

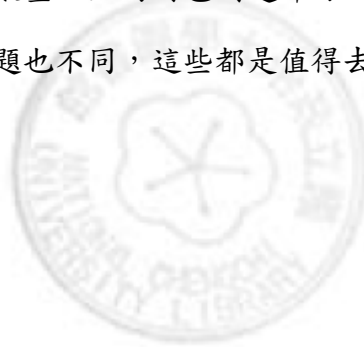
本論文提出一個提供彈性保護包的通訊協定協定，PR-TCP，我們提出的 PR-TCP 有兩個版本，Basic PR-TCP 及 Single-Side PR-TCP，Basic PR-TCP 依照上層需求將封包設定為不同的保護等級，在封包遺失時，依照不同的保護等級做不同的處理，能盡量確保重要封包的到達且不因不重要封包遺失浪費時間重傳，彈性更高，更減少額外的網路資源耗用，然而使用 Basic PR-TCP 必須在傳送端及接收端都裝上 PR-TCP，比較不容易推行，所以我們提出 Single-Side PR-TCP，只裝設在傳送端，接收端可以使用任何 TCP。Single-Side PR-TCP 依照上層需求將封包設定為不同的保護等級，並且使用傳送沒有內容的假封包方法，降低接收端因為沒收到封包而一直等待的機會，降低重傳的 overhead，適用於 client/server 架構。另外我們在 PR-TCP 上層加上自動重建遺失封包機制，降低封包錯誤的機會發生。

我們使用兩個 Video 影像作為傳輸標的，在高遺失率的有線與無線網路的環境中進行實驗。當傳輸時間限制很短時，(例如影像會議的應用)，在有線的環境中，Basic PR-TCP 比 TCP Reno、TFRC 最少增加約 18% 的可解畫面封包數，比 UDP、TFRC 及 TCP Reno 的 PSNR 值最少高出約 15%，比 TCP Reno 及 TFRC 最少節省了 12% 的頻寬資源，Single-Side PR-TCP 比 Basic PR-TCP 的 PSNR 值約低了 11%，額外耗用的頻寬約多出 10%。在無線的環境中，Basic PR-TCP 比 TCP Reno、TFRC 最少增加約 19% 的可解畫面封包數，比 UDP、TFRC 及 TCP Reno

的 PSNR 值最少高出約 20%，比 Single-Side PR-TCP、TCP Reno 及 TFRC 最少節省了 15% 的頻寬資源，Single-Side PR-TCP 比 Basic PR-TCP 的 PSNR 值約低了 14%。當傳輸時間限制較充裕時（例如 VoD 應用），Basic PR-TCP 雖然比 TCP Reno 及 TFRC 降低了約 3% 的 PSNR 值，但是最少能節省 8% 的頻寬耗費，Single-Side PR-TCP 的 PSNR 值跟 Basic PR-TCP 相近，但是額外耗用約 5% 的頻寬資源。我們的實驗顯示，在傳輸影音資訊這類不需完全保證封包的遞送時，Partial-Reliable TCP 可以得到較高的效能。觀察實際影像播放，畫面較清楚也較流暢。

我們提出的 Partial-Reliable TCP 比起 TCP、UDP 更適用於封包有不同重要性的即時資訊服務。

選擇性保證封包送達的想法還可以有很多延伸，例如保證特定的封包到達或是在一資料流裡只要一定數量以上的封包到達即可，另外 Protection Class 的不同，可以應用及解決的問題也不同，這些都是值得去思考的地方。



參考文獻

- [1] A.L. Caro, et al., “SCTP: A Proposed Standard for Robust Internet Data Transport,” *IEEE Comp.*, vol. 36, no. 11, pp. 56-63, Nov. 2003.
- [2] D. Clark, “Window and Acknowledgement Strategy in TCP,” *IETF RFC 813*, 1982.
- [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 no. 2, pp. 1-14, 1989.
- [4] K. Chandra and A.R. Reibman, “Modeling one and two layer variable bit rate video,” *IEEE/ACM Trans. on Networking*, vol. 7, no. 3, pp. 398–413, Jun. 1999.
- [5] W.C. Feng and J. Rexford, “Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video,” *IEEE Trans. on Multimedia*, vol. 1, no. 3, pp. 302–312, Sep. 1999.
- [6] S. Floyd and T. Henderson, “The NewReno Modification to TCP's Fast Recovery Algorithm,” *IETF RFC 2582*, 1999.
- [7] M. Handley, S. Floyd, J. Padhye, and J. Widmer, “TCP friendly rate control protocol specification (TFRC),” *RFC3448*, Jan. 2003.
- [8] V. Jacobson, “Congestion Avoidance and Control,” *Proc. of ACM SIGCOMM*, pp. 314-329, Aug. 1988.
- [9] V. Jacobson, “Modified TCP Congestion Avoidance Algorithm,” *IETF RFC 2001*, Apr. 1990.

- [10] E. Kohler, et al., “Designing DCCP: Congestion Control without Reliability,” ICNP 2003.
- [11] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid - A Framework for Video Transmission and Quality Evaluation,” *Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pp. 255-272, Sep. 2003.
- [12] Y.N. Lien, H.C. Jang, T.C. Tsai and H. Luh, “Budget Based QoS Management Infrastructure for All-IP Networks,” *Proc. of the IEEE 25th International Conference on Advanced Communication Technology (ICACT2005)*, Feb. 2005.
- [13] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgement Options,” *IETF RFC 2018*, 1996.
- [14] J. Postel, “Internet Protocol,” *RFC 760, USC/Information Sciences Institute*, 1980.
- [15] J. Postel, “Transmission Control Protocol,” *IETF RFC 793*, 1981.
- [16] J. Postel, “User Datagram Protocol,” *IETF RFC 768*, 1981.
- [17] M. Piecuch, K. French, G. Oprica, and M. Claypool, “A Selective Retransmission Protocol for Multimedia on the Internet,” *Proc. of SPIE Multimedia Systems and Applications*, Nov. 5-8, 2000
- [18] S. Ryu, C. Rump, and C. Qiao, “Advances in Internet Congestion Control,” *IEEE Communications Surveys and Tutorials*, vol 5. no. 2, 2003.
- [19] J. D. Salehi, Z. L. Zhang, J. F. Kurose, and D. Towsley, “Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing,” *IEEE/ACM Trans. on Networking*, vol. 6, no. 1, pp. 397–410, Aug. 1998.

- [20] S. Sen, J. L. Rexford, J. K. Dey, J. F. Kurose, and D. F. Towsley, "Online Smoothing of Variable-Bit-Rate Streaming Video," *IEEE Trans. on Multimedia*, vol. 2, no. 1, pp. 37–48, Mar. 2000.
- [21] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *IETF RFC 2001*, 1997.
- [22] J. Vieron and C. Guillemot, "Real-time constrained TCP-compatible rate control for video over the Internet," *IEEE Trans. on Multimedia*, vol. 6, no. 4, pp. 634–646, Aug. 2004.
- [23] M. Van and S. Shankar, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 50-58, Aug. 2005.
- [24] S. Wenger, "H.264/AVC over IP," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.
- [25] D. Wu, Y.T. Hou, W. Zhu, T.H. Chiang, Y.Q. Zhang, and H.J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 923-941, Sept. 2000.
- [26] D. Wu, Y.T. Hou, W. Zhu, and Y.Q. Zhang, "Streaming Video over the Internet: Approaches and Directions," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, Mar. 2001.
- [27] T. Wiegand, J. Gary, G. Bjontegaard, and A. Luthra, "Overview of the H.264 / AVC Video Coding Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no 7, pp. 560-576, July 2003.
- [28] G.K. Wallace, "The JPEG Sill Picture Compression Standard,"

- Communications of the ACM*, vol. 34, no. 1, pp. 31-44, Apr. 1991.
- [29] C.Y. Yu, C.H. Ke, C.K. Shieh, and N. Chilamkurti,
“MyEvalvid-NT - A Simulation Tool-set for Video Transmission
and Quality Evaluation,” *TENCON 2006.*, pp. 1-4, Nov. 2006.
- [30] 3rd Generation Partnership Project, “Technical Specification Group
Services and Systems Aspects: Architecture for an All IP network”,
3GPP TR 23.922 version 1.0.0, Oct. 1999.
- [31] “The Network Simulator,” <http://www.isi.edu/nsnam/ns/>.
- [32] <http://140.116.72.80/~smallko/ns2/ns2.htm>.
- [33] “YUV Video Sequences,” <http://trace.eas.asu.edu/yuv/index.html>.
- [34] ISO/IEC International Standard 11172; “Coding of moving pictures
and associated audio for digital storage media up to about 1.5
Mbits/s,” Nov. 1993.
- [35] ISO/IEC International Standard 13818; “Generic coding of moving
pictures and associated audio information,” Nov. 1994.