

## 第四章 AARC 運用於即時監視系統

視訊串流(Video Streaming)連續地在有限資源的網路環境中傳輸影像，影像品質往往受限於接收端的解碼能力強弱與頻寬大小[13]，一旦電腦裝置的 CPU 速度不夠快，代表其所能接收到的串流服務有限，也容易造成過久的 CPU 運算時間，所以假如 IP-CAM 能適時偵測接收端的特性，然後給予適當的影像串流，將有助於滿足不同使用者的要求。現列舉範例如圖 4.1 所示：

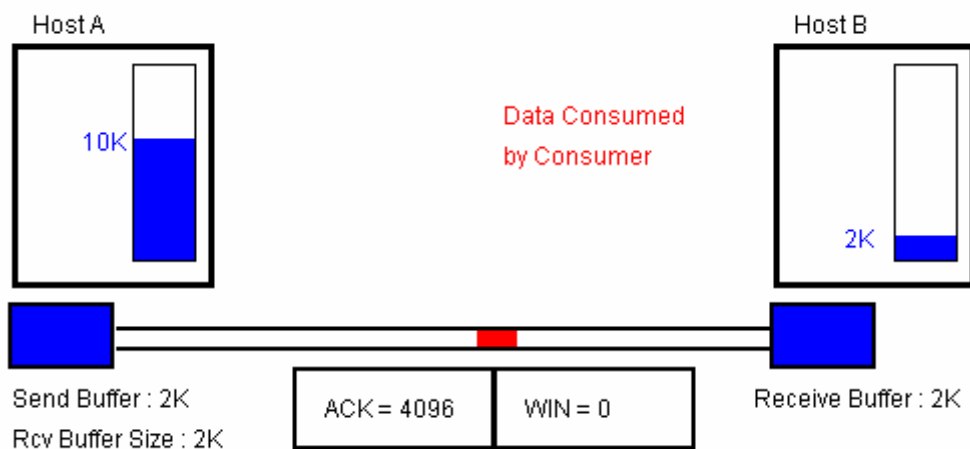


圖 4.1 TCP Buffer 之擁塞示意圖(摘錄自 [20]所列 Java Applet 網頁)

Host A 將緩衝區的資料持續送往 Send Buffer，由於 Host B 的 CPU 速度較慢，來不及消化緩衝區內的資料，因此位於 TCP Buffer 內部的 Send Buffer 與 Receive Buffer 形成資料滿溢(Overflow)[10]，最後造成

Host A 的緩衝區必須拋棄資料與 Host B 的資料停頓。在此章節中，我們將介紹自行設計的 AARC 機制及其解決方法來說明，並各自針對其方法功能及特性做介紹。

#### 4.1 主動可調適資源控制(AARC)方法簡介

在討論完簡易使用 IP-CAM 的方法後，本章節將焦點放在如何調整 IP-CAM 的影像串流，來提高使用者端 CPU 解碼影像的成功率，以提供適合不同使用者裝置的即時影像服務。由於即時監視系統是屬於單一播送(Unicast)的視訊串流，可以針對特定用戶來做服務，與群播(Multicast)的視訊串流有所差別，因為群播機制只能給予統一的服務項目[14]，所以點對點(End-to-End)的網路架構就非常適合本文所設計的主動可調適資源控制(AARC)機制，針對用戶的特性來提供即時影像。

先前的研究中，對於處理即時影像的問題，說明丟棄(Drop)部分影像將可有效而簡單地解決網路頻寬不足與接收端的運算瓶頸[15]，該如何選擇不必要的影像也被討論[16]。本文選擇容易實作又不影響即時影像的效率，也不破壞原始影像的品質之下，期望能藉由丟棄部分的影像畫面(如 P Frame)來增加接收端的處理效能，因此提出主動可調

適資源控制(Active-Adaptive Resource Control, AARC)機制，主動(Active)地監視使用者端的 CPU 效能，將收集的資訊回報給 IP-CAM 來傳輸可調適(Adaptive)的即時串流，在此資源(Resource)代表著控制影像畫面數目與改變相對應的網路頻寬。

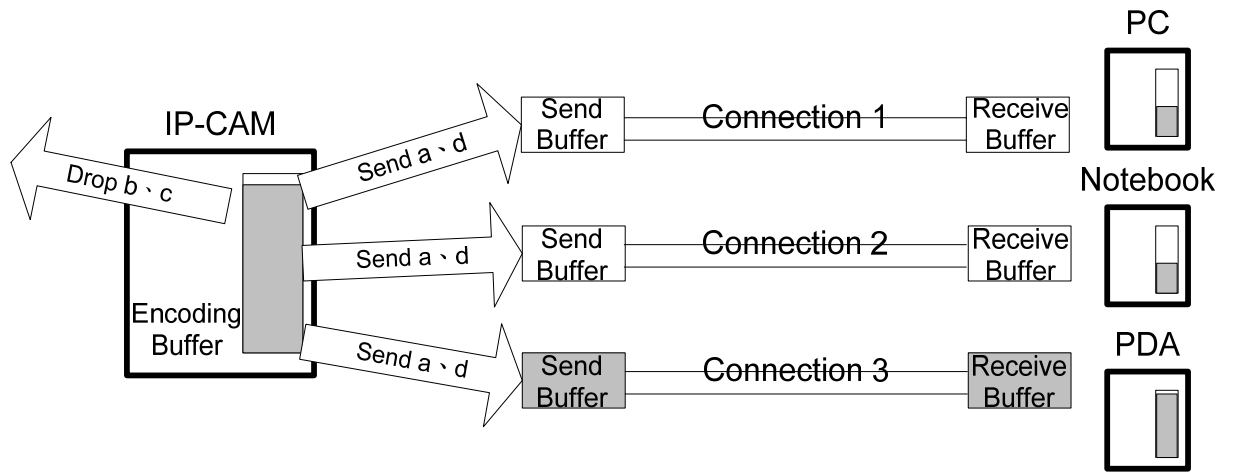


圖 4.2 IP-CAM drop frame 影響傳輸資料之示意圖

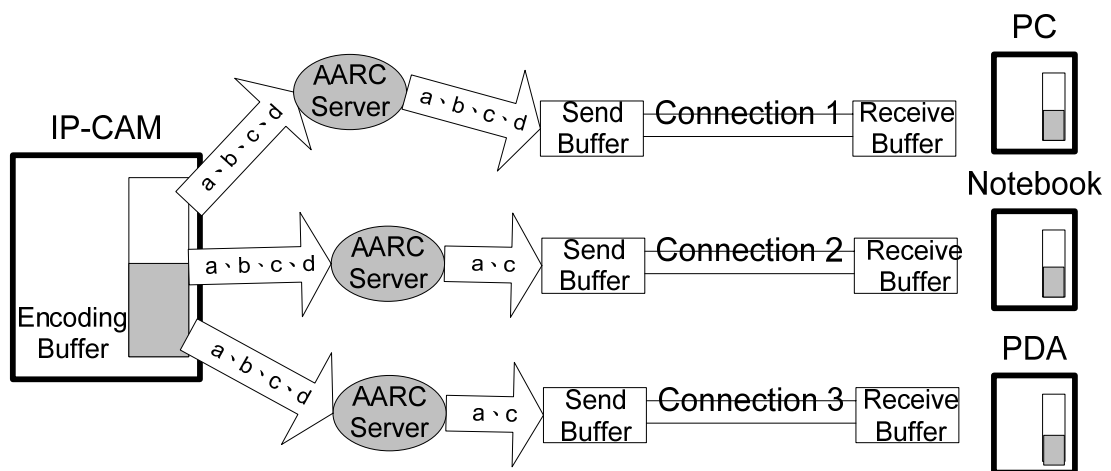


圖 4.3 SIP-CAM 之 AARC Server 影響傳輸資料之示意圖

圖 4.2 表示 IP-CAM 一旦有效能較差之 CPU 連線時，因為一連串的 TCP Buffer 阻塞，所以造成 IP-CAM 丟棄畫面 b 和畫面 d，而讓 CPU 速度快的 PC 連帶拖累而只能收到畫面 a 和畫面 d。如果圖 4.3 的 SIP-CAM 加入 AARC Server 就能使用畫面處理器，讓 PDA 只能收到畫面 a 和畫面 c，但是 PC 則完整收到畫面 a、b、c、d。

AARC 機制主要有兩個元件與一個控制方法，首先 AARC Server 扮演著訊息接收端與資料傳送端的角色，設定在 SIP-CAM 內部；而 AARC Client 則是訊息傳送端與資料接收端的角色，設定在 MPlayer 裡面，最後透過四種等級的控制訊號來切換不同的情況，如圖 4.4 所示。

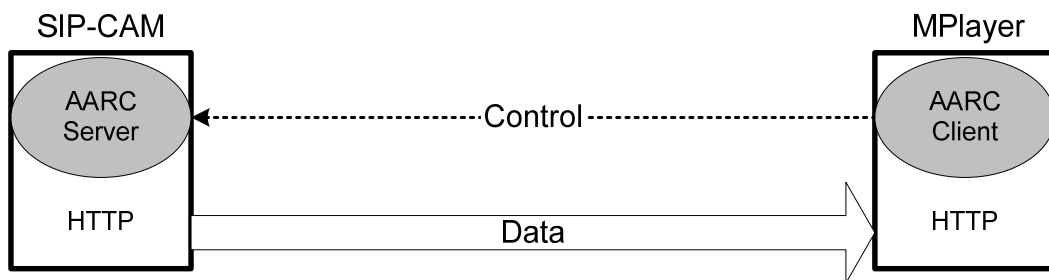


圖 4.4 AARC Server 與 AARC Client 間通訊表示圖

由圖 4.5 可知，AARC Server 存在 HTTP 之中，主動監控 MPEG-4 多媒體進入 TCP/IP 前的畫面，首先藉由 AARC 機制所設定的兩種控

制介面，模式 A 與模式 B(如圖 4.5 之步驟五虛線所標記)傳遞四種等級的控制訊號，改變步驟一、二、四的資料數量。而步驟三的資訊將在模式 B 啟動後，使用封包伺服器(Packet Server)擷取 TCP 封包，用來觀察網路流通的訊息，如 TCP 封包檔頭欄位之 Window size，可以藉此數據來衡量 Client 端的接收緩衝區容量狀況。

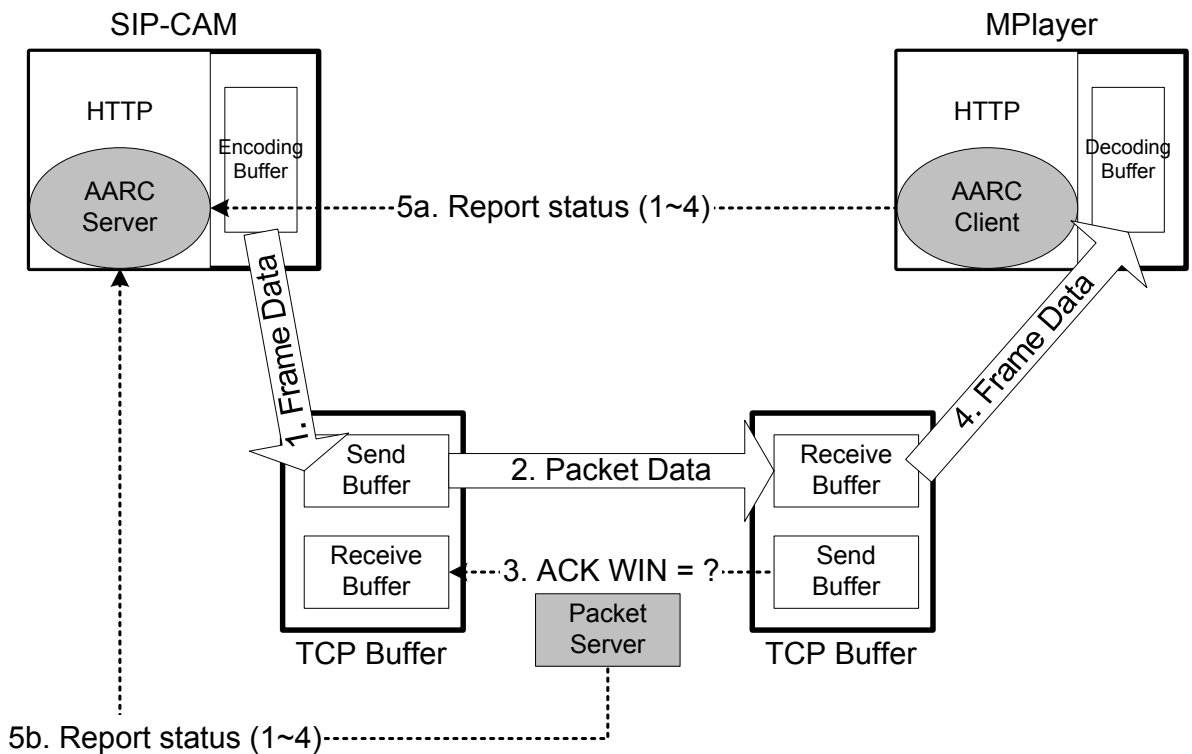


圖 4.5 AARC 運用於即時監視系統之方法結構圖

以下將循序說明圖 4.5 所有步驟：

(I) Frame Data：SIP-CAM 將 Encoding Buffer 內的 MPEG4-ES 畫

面串流，傳遞至 TCP Buffer 的 Send Buffer。

(II) Packet Data: SIP-CAM 的 Send Buffer 再以 Packet 的資料形式，傳輸至 Client 端的 TCP Buffer 其中之一的 Receive Buffer。

(III) ACK WIN=? : 第三步驟 Client 端的 TCP Buffer 另一部份之 Send Buffer 回覆 ACK 給 SIP-CAM 的 Receive Buffer。

(IV) Frame Data : 隨即 Client 端的 Receive Buffer 將資料串流傳遞至 Decoding Buffer 準備解碼。

(V) a. Report status (1~4) : Client 端將檢視解碼過程中，Decoding Buffer 的剩餘容量，接著比對模式 A 的期望門檻值，並將對應到之狀態數值向 AARC Server 回報。

(VI) b. Report status (1~4) : 在 SIP-CAM 中將有一個用作撈取第三步驟中封包的 Packet Server，用來檢視 Client 端的 Receive Buffer 剩餘容量，接著比對模式 B 的期望門檻值，並將對應到之狀態數值向 AARC Server 回報。

## 4.2 AARC 主要模式與流程

本文所設計的 AARC 機制分成模式 A 與模式 B，首先如果 Client 端具有 AARC 機制的控制介面，將可以對 IP-CAM 進行訊息控制，此

即模式 A；但是如果 Client 端不懂得 AARC 機制或是控制介面失效，將轉換成模式 B，亦即具有 AARC 機制的 IP-CAM 將可自行實施 AARC 機制運算。

(I) 模式 A (Client 端軟體控制模式)：Client 端中的 AARC Client 將 CPU 的解碼緩衝區(Decoding Buffer)剩餘容量，比對下述圖 4.6 的期望門檻值之後，將對應結果回報給 AARC 機制中的 AARC Server。

$$Level (Decoding Buffer) = \begin{cases} 1, & 75\% \leq size < 100\% \\ 2, & 50\% \leq size < 75\% \\ 3, & 25\% \leq size < 50\% \\ 4, & 0\% \leq size < 25\% \end{cases} \dots\dots \text{公式(一)}$$

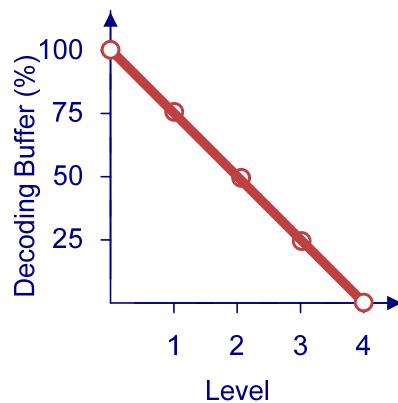


圖 4.6 模式 A 之期望門檻值座標圖

(II) 模式 B (IP-CAM 端緩衝區控制模式): Packet Server 監視著 TCP Buffer 之中 Receive Buffer 的剩餘容量，比對下述圖 4.7 的期望門檻值之後，將對應結果回報給 AARC 機制中的 AARC Server。由於 TCP Buffer 一旦 Overflow 就會造成網路擁塞，所以期望門檻值以倍數遞減，避免在臨界點 100% 附近停留。

$$Level (Client Receive Buffer) = \begin{cases} 1, & 90\% \leq size < 100\% \\ 2, & 70\% \leq size < 90\% \\ 3, & 40\% \leq size < 70\% \\ 4, & 0\% \leq size < 40\% \end{cases} \dots\dots \text{公式(二)}$$

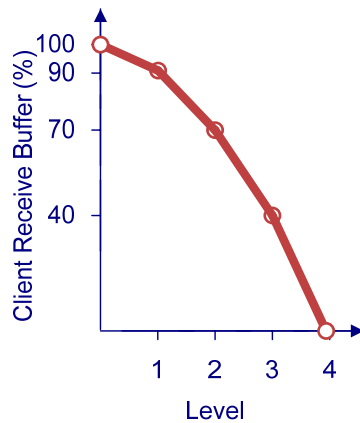


圖 4.7 模式 B 之期望門檻值座標圖

AARC 機制將會依照四級狀態機而依序改變狀態，如圖 4.8 所示，並且對應到之等級將啟動畫面處理器，如此循序的跳動狀態是為維持



影像順利播放的原則。

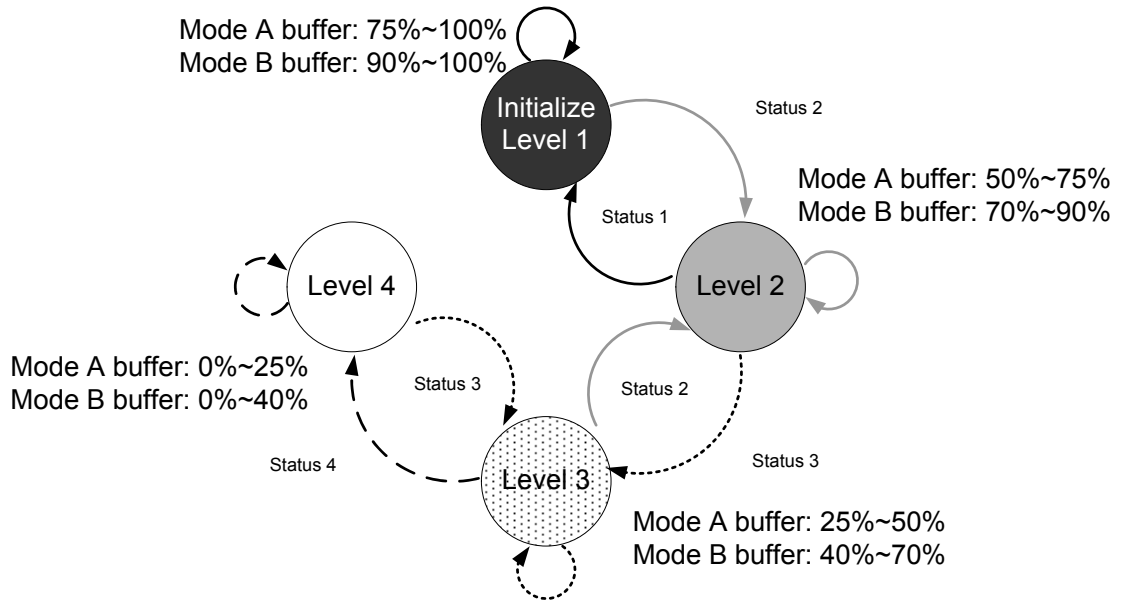


圖 4.8 模式 A 與模式 B 之四級狀態圖

### 4.3 AARC 之畫面處理器

畫面丟棄濾波器((Frame-Dropping Filters)藉由調整影像畫面的數目(增加或是減少 I Frame 與 P Frame 的數目)，用以穩定網路資源的分配，在網路中是最有效而為大家所使用的[15]。

AARC 機制目前只設計了四種具代表性的畫面處理器演算法，如圖 4.9 所示，可依照使用者的對應門檻值來選擇[10]：

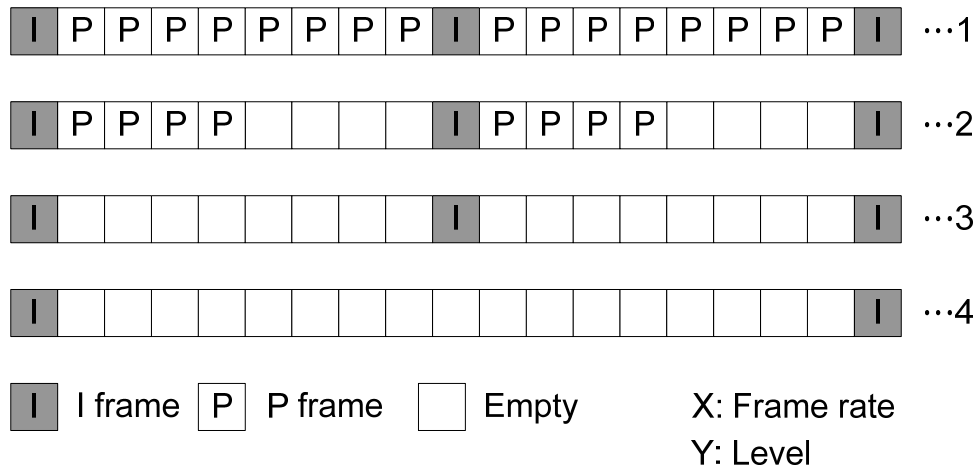


圖 4.9 四種畫面處理器之示意圖

(I) Level 1 (Keep all Frame)：原本 IP-CAM 所提供的 I Frame 與 P Frame，沒有丟棄任何畫面，所以影像品質最佳，而且畫面傳輸率為 30fps(frame per second)。

(II) Level 2 (Skip some P Frame)：經由 Level 1，丟棄部分的 P Frame，其畫面傳輸率為 18fps，擬程式碼如下：

```

Compute Queue Occupation Level of Sender Application Queue
If Queue Occupation Level > Occupation Level Threshold
  Starting at queue head to search first P frame
  While (not I frame)
    If (Drop half block of P frame is enabled)
      Discard frame
    Drop half block of P frame is disabled
  end
else
  Drop half block of P frame is enabled
    
```

```

        end
      Next frame
    end
  end
end

```

(III) Level 3 (Skip all P Frame): 經由 Level 2, 丟棄所有的 P Frame,

其畫面傳輸率為 4fps, 擬程式碼如下:

```

Compute Queue Occupation Level of Sender Application Queue
If Queue Occupation Level > Occupation Level Threshold
  Starting at queue head to search first P frame
  While (not I frame)
    Discard frame
    Next frame
  end
end
end

```

(IV) Level 4 (Skip all P Frame & some I Frame): 經由 Level 3, 丟棄

部分的 I Frame, 畫面傳輸率為 2fps, 擬程式碼如下:

```

Compute Queue Occupation Level of Sender Application Queue
If Queue Occupation Level > Occupation Level Threshold
  Starting at queue head to search first P frame
  While (not I frame)
    Discard frame
    Next frame
  end
  If (Drop I frame is enabled)
    Discard I frame
  Drop I frame is disabled
end
else

```

```
Drop I frame is enabled
end
end
```

本文設計 AARC 機制之畫面處理器演算法的概念，主要由 Rate Shaping 中的畫面丟棄濾波器而來，其好處加諸於即時監視系統有[8]：

- (I)改善畫面品質：因為畫面處理器瞭解多媒體串流的編碼格式，當視訊串流遇到網路擁塞情況時，可以溫和地降低畫面品質來丟棄封包，而不會破壞視訊串流的畫面。
- (II)增加頻寬效能：畫面處理器可藉由丟掉延遲過久或已經破壞的畫面來節省網路頻寬，特別是在高階 CPU 的接收端環境。
- (III)減少 CPU 負擔：CPU 只需要運算較少的影像位元率，將可以增加其他工作項目與減少系統資源支出，並且降低功率消耗，延長手持式裝置的使用時間。
- (IV)降低影像延遲：畫面處理器可以減少龐大資料的傳送，使得高階 CPU 的接收端可以盡量收到即時的影像畫面。