

# 兼顧即時性及品質保證之終端影音串流傳輸機制

## A Low-delay QoS Transmission Mechanism for Mobile Multimedia

周業凱      施啟煌      鄭雯方      林孟平  
Yeh-Kai Chou, Chi-Huang Shih, Wen-Fang Cheng, Meng-Ping Lin

### 中文摘要

本論文提出一個兼顧即時性及品質保證之終端影音設備的調適品質傳輸機制，係採用具交錯編碼與循序傳輸的前向糾錯編碼傳輸方法，目的在於解決終端影音傳輸的即時性以及行動網路下品質不佳的問題。現有的通道編碼技術雖已針對無線網路的資料傳輸具保護作用，然而在變化劇烈且頻寬有限的行動網路環境下仍需面對多種網路傳輸挑戰，包括網路壅塞與訊號錯誤等所導致的綜合性問題：資料回復效能不佳以及延遲過高仍是通道編碼的主要缺陷之一。經由實驗分析，相較於現有的技術，所提的方法能以交錯編碼與循序傳輸的結構適應於無線傳輸環境，且同時兼顧以下兩個目標：(1)低延遲傳輸結構，保有即時性；(2)良好的錯誤回復效果，影音品質有保證。

### Abstract

This paper presents a low-delay QoS transmission mechanism using FEC interleaving coding and sequential transmission (ICST) for mobile devices. In a general wireless transmission environment, challenges arise when traditional transmission mechanisms need to deal with both network loss and real-time issue under limited bandwidth constraint. Therefore, the proposed mechanism aims to achieve low-delay transmission while maintaining the high data recovery efficiency. Compared with the traditional mechanisms, the simulation results demonstrate that the proposed mechanism can achieve the same data recovery efficiency by means of using interleaved FEC coding. Furthermore, the delay performance can be improved based on the sequential transmission approach. As a result, the proposed mechanism has two main contributions: (1) the low-delay transmission for real-time applications; and (2) the effective data recovery to combat burst packet losses.

### 關鍵詞(Key Words)

低延遲 (low delay)  
品質保證 (quality of service ; QoS)  
向前糾錯碼 (forward error correction ; FEC)  
交錯編碼 (interleaving coding)  
循序傳輸 (sequential transmission)  
行動終端 (mobile device)

## 1 · 前言

隨著智慧終端裝置越發普及，使用智慧裝置進行影音分享及觀看的系統也越來越多，根據經濟部數位內容產業推動辦公室表示，數位匯流結合行動通訊軟體與動畫、遊戲、影音等功能，將創造新興消費商機[1]。據2014通訊產業年鑑，個人行動裝置的產值在2013年達到149億美元，而2014總產值將較2013成長3.7%，顯示裝置產值趨勢仍持續上揚。加上新興的數位匯流服務，未來行動使用者將可接收各類由電信、網路與電視等來源的多樣化資料。據統計，台灣使用4G服務的民眾，最常使用的服務包括即時通(83.2%)、線上影片(69.9%)、手機遊戲(60%)、內容上傳(56.2%)及線上音樂(52.5%)等。Google行動網路及使用者行為調查報告也顯示出90%的智慧型手機用戶會透過手機觀賞影片，且有31%的使用者每天至少使用1次影片功能[2]。

使用智慧終端進行影音資料上傳或串流分享意味著將使用行動網路，目前雖已正式進行4G時代，但行動網路仍常因使用人數較多而造成頻寬受限，或因無線傳輸技術引起頻寬於短時間內劇烈變化，其結果將使得使用者無法取得完整資料來進行正確的資訊顯示。由於現今的網路並不具備品質保證(quality of service; QoS)的機制，因此多採取通道編碼(channel coding)的技術以低延遲的方式提供可靠性的資料傳輸。典型的通道編碼應用實例包括了無線通訊協定如802.11 Wireless LAN [3]與802.16 WiMAX [4]，以及影音串流服務如視訊會議等等。通道編碼的核心技術為向前糾錯碼(forward error correction; FEC)，在傳送端將原始資料編碼出FEC資料再將兩者一併傳送，接收端則根據資料接收的狀況，在可容忍資料錯誤數量內執行解碼動作以完全回復原始資料[5]。儘管FEC的使用提供了錯誤回復的功能，但需要付出增加頻寬使用量的代價，且由於FEC編碼結構與傳輸機制的關係，也間接造成延遲的問題[6-10]。綜合FEC應用在網路傳輸的優劣分析，對智慧

終端影音串流分享於行動網路應用而言，設計一個低延遲且品質保證的機制，才能有效提升行動影音服務之分享品質。

本論文提出一個兼顧即時性及品質保證之傳輸方法，採用交錯編碼及循序傳輸的FEC編碼機制，結合到無線行動網路傳輸系統，可以有效對抗一般及突發封包遺失兩種資料遺失狀況，且相對於之前的交錯編碼更有效提升其應用之即時性，因此同時兼顧了即時性以及資料回復效能。

本論文架構簡介如下：第二章說明傳統循序編碼及交錯傳輸結構的FEC機制；第三章說明本論文提出之方法，即交錯編碼及循序傳輸之機制；第四章為實驗結果；本論文的相關結論整理於第五章。

## 2 · 傳統FEC交錯傳輸機制

本節首先簡介傳統FEC交錯傳輸機制，說明傳統機制採用循序編碼及交錯傳輸(Sequential Coding and Interleaving Transmission; SCIT)的架構，並使用數學方法建立延遲分析模型來觀察傳統機制的延遲效應。

為了便於說明，本論文假設FEC編碼演算法使用Reed-Solomon(RS)碼，以網路封包為編碼單位，因此 $k$ 個原始封包可編碼出 $h$ 個FEC封包，此 $(k+h)$ 個封包稱為一個FEC區塊。FEC區塊中，只要收到任 $k$ 個封包即可解碼出所有原始封包；換句話說，FEC區塊中最大可容忍的封包遺失或錯誤數量為 $h$ 。上述即為標準的FEC機制，為了與具交錯功能的FEC有所區別，在本論文中將其通稱為非交錯型FEC(non-interleaving FEC)。

### 2.1 機制簡介

傳統FEC交錯傳輸機制的基本作法是將 $d$ 個FEC區塊的封包平均打散傳送，因此封包的傳送會依序把每個區塊的第一個封包都傳送完畢之後，接著依序傳送每個區塊的第二個封包，依此類推，一直到將每個區塊的最後一個封包(也就是第 $(k+h)$ 個封包)都傳送出去為止。參與

交錯傳輸的區塊數量  $d$  即為交錯階層 (interleaving degree)。

圖 1 顯示了一個傳統 FEC 交錯傳輸機制的例子。此例中假設 FEC 區塊的大小為 4—包括 3 個原始封包以及 1 個 FEC 編碼封包，並且交錯階層為 3。在圖一中，白色區塊示意為原始封包，灰階斜線區塊為 FEC 編碼封包。原始封包依序編號為 1~9，FEC 編碼封包則依序編號為 1~3。

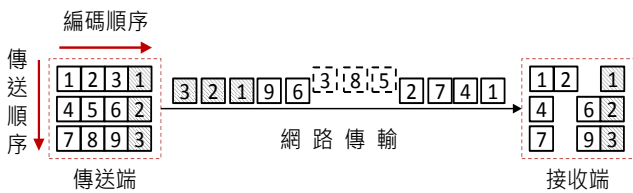


圖 1 傳統 FEC 交錯傳輸機制示意圖

如圖 1 所示，9 個原始封包構成三個 FEC 區塊：(1,2,3) · (4,5,6) · (7,8,9)。每個 FEC 區塊各自編碼出自己的 FEC 封包。傳送端針對此三個區塊採用交錯傳輸的方法，因此 9 個原始封包的傳輸順序為：1,4,7,2,5,8,3,6,9。假設傳輸過程中發生的連續性封包遺失量是 3，因此封包 (5,8,3) 無法被接收端順利接收。假如沒有交錯傳輸的機制，此連續封包遺失會造成某一個 FEC 區塊無法回復，畢竟每個 FEC 區塊最多只可回復 1 個封包遺失。但在圖 1 的示意範例中，由於採用交錯傳輸，此三個連續封包遺失會被分散到三個不同的 FEC 區塊裡，如接收端所示，三個區塊可以進行 FEC 解碼，使得 9 個原始封包都可以成功被接收端所使用。由於圖 1 所示的傳統 FEC 交錯傳輸機制，是依照封包編號進行 FEC 編碼，並依照 FEC 區塊順序做交錯傳輸，因此在本論文中，將傳統機制稱為 SCIT (Sequential Coding and Interleaving Transmission; 循序編碼交錯傳輸) FEC。

為了進行交錯傳輸，SCIT FEC 必須在傳送端與接收端分別維護一個封包緩衝區 (buffer)。傳送端必須等到交錯階層的所有區塊封包都到齊了，才能進行 FEC 編碼以及交錯傳輸。而接收端也因為封包接收的順序並不是按照原始編

號，需要等待同個 FEC 區塊的所有封包到齊方能進行 FEC 解碼。依照以上所述，SCIT FEC 儘管能夠有效處理連續性封包遺失的問題，但另一方面卻因為緩衝區的等待延遲而增加了傳輸所需的時間。更甚者，當網路環境的連續封包遺失數量更多的時候，交錯階層的數量也要隨之增加，就會造成更長的延遲時間！較長的延遲時間可能有資料接收不連續的問題，對網路應用程式造成使用者經驗不佳的負面影響 [6]。

## 2.2 延遲時間分析

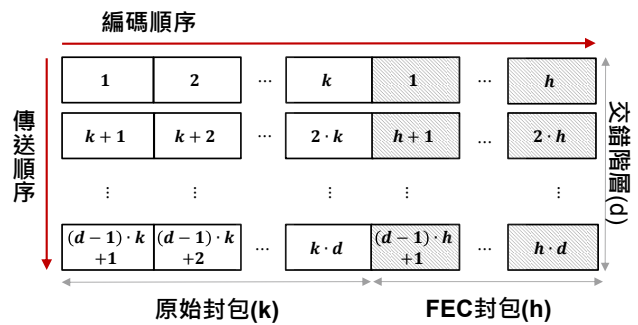


圖 2 SCIT FEC 的延遲時間計算示意圖

本節介紹 SCIT FEC 的延遲計算模型。首先定義封包間隔時間 (packet inter-arrival time) 為  $T_{in}$ ，表示系統產生兩個封包的時間間隔，而傳送端傳送一個封包所需時間為  $T_{out}$ 。在本篇論文中，我們採用的是最大延遲時間的計算，也就是假設第一個封包會在傳輸中發生遺失，此最大延遲的計算區間即為第一個封包進入傳送端緩衝區的時間到此封包在接收端被成功解碼的時間，並定義此 SCIT FEC 的最大延遲為  $D_{SCIT}$  (Delay-time for SCIT)。

圖 2 為我們計算延遲所使用的 FEC 區塊結構圖：每個 FEC 區塊有  $k$  個原始封包、 $h$  個 FEC 封包，交錯階層為  $d$ 。根據前述  $D_{SCIT}$  的定義， $D_{SCIT}$  包含了三個部分：(1) 第一個封包到達之後，在傳送端等待其他  $(k \times d - 1)$  原始封包到達封包緩衝區的時間；(2) 傳送端傳送  $(k \times d)$  個原始封包以及  $((h - 1) \times d + 1)$  個 FEC 封包的時間；(3) 封包在網

路的傳遞時間 $T_{prog}$ 。在(2)的部分，由於是計算最大延遲，在接收端解碼第一個封包需要收到第一個區塊的所有封包，但因為SCIT FEC採用交錯傳輸，所以要完整收到第 $h$ 個第一區塊的FEC封包前，會先接收所有 $d$ 個區塊的 $(h-1)$ 個FEC封包，所以FEC封包的傳送輸量為 $(h-1) \times d + 1$ ： $(h-1) \times d$ 是所有 $d$ 個區塊的 $(h-1)$ 個FEC封包數量，最後加上第一個區塊的第 $h$ 個FEC封包。因此， $D_{SCIT}$ 可寫為：

$$D_{SCIT} = (k \cdot d - 1) \cdot T_{in} + (k \cdot d + (h - 1) \cdot d + 1) \cdot T_{out} + T_{prog} \quad (1)$$

以圖1的例子來說，套用公式(1)的定義可計算出其 $D_{SCIT}$ 為 $8 \times T_{in} + 10 \times T_{out} + T_{prog}$ ，其中 $(T_{in}, T_{out}, T_{prog})$ 是系統參數。

### 3 · 低延遲的FEC交錯傳輸機制

本節首先簡介本論文所提出的FEC交錯傳輸機制，說明此機制採用的交錯編碼及循序傳輸 (Interleaving Coding and Sequential Transmission; ICST) 架構，並使用數學方法建立延遲分析模型來跟2.2節的傳統機制模型進行數據比較。

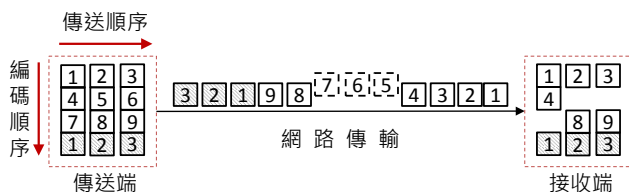


圖 3 ICST FEC交錯傳輸機制示意圖

#### 3.1 機制簡介

為了解決傳統交錯機制的時間延遲問題，本論文所提ICST FEC，主要目標在於降低交錯傳輸的延遲，並且能維持交錯傳輸的既有效能。圖3說明了使用ICST FEC的例子，所用參數與圖1示皆沿用圖1的定義。如圖3所示，封包的傳送順序和封包編號相同，這意謂當應用程式產生一原始封包的時候，該封包即可立刻被傳送出去；

同時，傳送端會保有一份該原始封包的備份。當9個原始封包都被傳送完成時，這些原始封包的備份便可依交錯順序構成三個FEC區塊： $(1,4,7) \cdot (2,5,8) \cdot (3,6,9)$ 。此三個區塊各自進行FEC編碼，並隨後將三個區塊的FEC封包傳送出去。同樣地，假設與圖1發生連續性封包遺失的位置相同，此時封包 $(5,6,7)$ 無法到接收端。看似連續編號的封包，其實各自對應到不同的FEC區塊，因此實際上每一個區塊分別遺失一個封包，最終可以回復所有的原始封包。

比較圖1與圖3，ICST FEC與SCIT FEC同樣需有傳送端和接收端的封包緩衝區空間，但ICST FEC的編碼與傳送順序正好與SCIT FEC相反。ICST FEC的交錯機制在組成編碼區塊時進行，因此能達到交錯傳輸的效果。此外，ICST FEC是在應用程式封包到達時立刻傳送，因此傳送端並不需要花費額外的等待時間來收集足夠形成交錯階層所需的封包數量。除此之外，由於收取封包時也是按照封包編號順序接收，所以ICST FEC的接收端也可以較快的將收到的封包呈送給應用程式進行處理。舉例來說，圖1的前四個接收封包為 $(1,4,7,2)$ ，而圖3的前四個接收封包為 $(1,2,3,4)$ ，因而圖3的ICST FEC已經可以直接送交 $(1,2,3,4)$ 給應用程式使用，但圖1的SCIT FEC則需要再等待6個封包的時間，直到接收端收到第一個區塊的FEC封包方能回復遺失的封包3，隨後送交 $(1,2,3,4)$ 給應用程式使用。簡要的說，ICST FEC仍舊保有處理連續性封包遺失的能力，並且具有較低的時間延遲。

#### 3.2 延遲時間分析

沿用2.2節計算SCIT FEC延遲時間的數學模型定義，本節介紹ICST FEC的延遲計算模型。圖4為我們計算延遲所使用的FEC區塊結構圖。ICST FEC的延遲時間( $D_{ICST}$ )可分為三個部分：(1)第一個封包到達之後，在傳送端等待其他 $(k \times d - 1)$ 原始封包到達封包緩衝區的時間；(2)傳送端傳送 $(k \times d)$ 個原始封包以及 $((h-1) \times d + 1)$ 個FEC封包的時間；(3)封包在網路的傳遞時間 $T_{prog}$ 。在(2)的部分，ICST FEC在原始封包產生



後便立刻傳送該封包；對於現今的網路應用來說，為了避免資料緩衝區溢出(buffer overflow)而造成資料遺失的現象，網路傳輸速率必須要大於應用程式資料的產生速率，因而傳送時間( $T_{out}$ )會小於下一個封包的間隔時間( $T_{in}$ )。根據以上觀察，在 $T_{in}$ 跟 $T_{out}$ 同時進行的狀況下， $T_{out}$ 的時間會被包含在 $T_{in}$ 時間裡頭，因而除了最後一個原始封包之外，其餘原始封包的傳送時間計算上便只會留下時間較長的 $T_{in}$ 。因此需要計算的FEC封包傳送輸量為： $(h-1) \times d + 2$ ； $(h-1) \times d$ 是傳送第一個區塊的第 $h$ 個FEC封包前的所有FEC封包，接下來的2分別是最後一個原始封包以及第一個區塊的第 $h$ 個FEC封包。

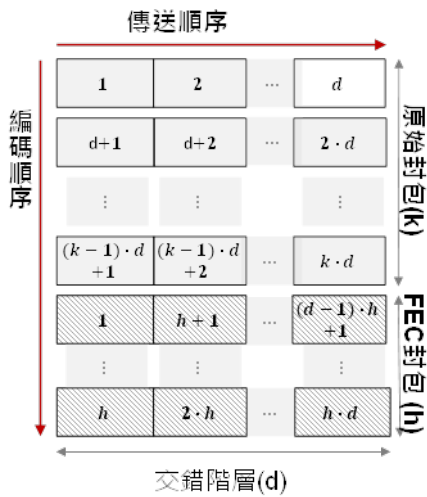


圖 4 ICST FEC的延遲時間計算示意圖

最後， $D_{ICST}$ 可表達為：

$$D_{ICST} = (k \cdot d - 1) \cdot T_{in} + ((h - 1) \cdot d + 2) \cdot T_{out} + T_{prop} \quad (2)$$

以圖3的例子來說，套用公式(2)的定義可計算出其 $D_{ICST}$ 為 $8 \times T_{in} + 2 \times T_{out} + T_{prop}$ ，其中( $T_{in}$ ,  $T_{out}$ ,  $T_{prop}$ )是系統參數。

## 4. 實驗結果與效能分析

### 4.1 延遲時間分析

本節以數值方法分析公式(1)跟公式(2)的差異。根據網路傳輸速率必須要大於應用程式

資料產生速率的事實，我們假設應用程式產生 $k$ 個原始封包與傳送 $n$ 個封包(包括 $k$ 個原始封包與 $h$ 個FEC封包)的關係式如下：

$$k \cdot d \cdot T_{in} \geq n \cdot d \cdot T_{out} \quad (3)$$

稍加整理我們可以得到：

$$\frac{T_{in}}{T_{out}} \geq \frac{n}{k} \quad (4)$$

接著，我們假定ICST FEC跟SCIT FEC的延遲時間不包含網路的傳遞時間 $T_{prog}$ ，因此兩者的時間比值是

$$\text{Delay Enhancement} = \frac{D_{SCIT} - T_{prog}}{D_{ICST} - T_{prog}} \quad (5)$$

表 1 SCIT FEC跟ICST FEC的延遲比較

$\frac{T_{in}}{T_{out}}$	Delay Enhancement ( $\frac{D_{SCIT} - D_{prog}}{D_{ICST} - D_{prog}} = \frac{8 \cdot T_{in} + 10 \cdot T_{out}}{8 \cdot T_{in} + 2 \cdot T_{out}}$ )
4	$\frac{62}{38} = 1.63$
3	$\frac{26}{18} = 1.44$
2	$\frac{42}{34} = 1.24$

以圖1跟圖3為例，參數群( $k$ ,  $h$ ,  $d$ )值為(3,1,3)。在假設公式(4)等號成立的情況下，我們可以計算出公式(5)的數值，如表1所示。由此可知， $T_{in}$ 跟 $T_{out}$ 越相近的話，ICST FEC所提供的時間效能改善越明顯；換句話說，ICST FEC較適用於資料較大的應用。對照公式(1)跟公式(2)，這個結果是合理的，因為小量資料的應用程式 $T_{in}$ 較大，造成延遲時間都集中在等待封包間隔時間，即便ICST FEC立即傳輸了原始封包，所節省的時間也比較有限。

### 4.2 模擬結果

為了瞭解ICST FEC跟SCIT FE兩者在實際

網路環境中的效能比較，本論文使用 NS-2(Network Simulator 2)模擬工具進行實驗。實驗環境的設定如下：資料傳送端以乙太網路與無線基地台連接，而資料接收端透過無線基地台接收傳送端的資料。網路頻寬是10 Mbps，傳輸延遲時間為10毫秒。傳送端上運作一個能輸出固定位元速率(constant bit rate; CBR)的 H.264影片”Foreman”，其輸出速率為3 Mbps。網路傳輸過程中由雙狀態馬可夫模型(two-state Markov model)產生封包遺失狀況。本實驗觀察三組FEC機制，分別為ICST FEC、SCIT FEC以及non-interleaving FEC(即不具交錯功能的標準FEC機制)；所有FEC機制均採用RS編碼，FEC區塊大小是15個原始封包與5個FEC編碼封包，交錯階層為3。

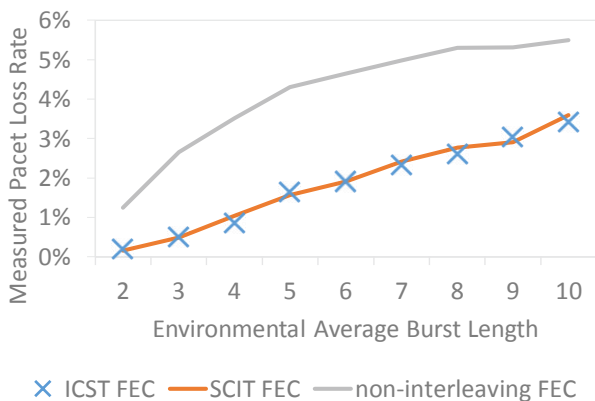


圖 5 不同連續封包遺失長度下，接收端所量測到的封包遺失率

第一個實驗中，網路的封包遺失率固定為6.5%，平均封包連續遺失長度(average burst length; ABL)設定為2~10個封包不等，三種不同FEC機制所得到的受測封包遺失率(measured packet loss rate)以及FEC延遲時間(FEC Delay)分別顯示於圖5與圖6。圖5的受測封包遺失率意謂經由接收端的FEC解碼之後所計算得來的封包遺失率，主要用來觀察FEC機制的資料回復效能。如圖5所示，隨著ABL增加，所有的FEC機制均會得到較差的資料回復結果，也就是較高的受測封包遺失率。從圖5我們也可以觀察到：

使用交錯傳輸可以增加FEC對抗連續封包遺失的影響，並且ICST FEC與SCIT FEC的效能是相同的。圖6的延遲時間所量

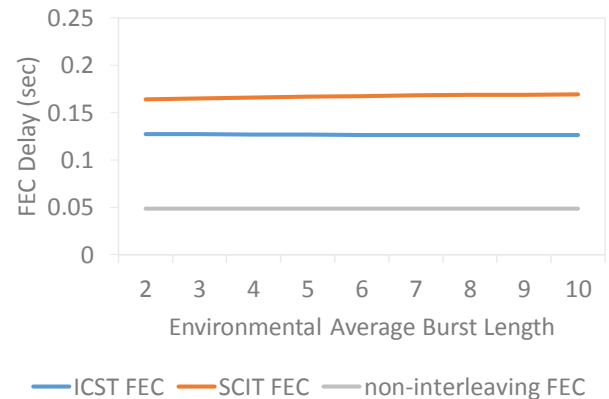


圖 6 不同連續封包遺失長度下，接收端所量測到的延遲時間

測的是端點到端點的時間，包含FEC的緩衝時間以及網路傳輸延遲。如圖6所示，non-interleaving FEC因為不需要交錯傳輸所需的資料緩衝，只需0.05秒的端點到端點延遲，傳統的SCIT FEC則觀察到約0.16秒的延遲，而我們所提出的ICST FEC則可以進一步降低延遲時間到0.12秒。以目前的實驗設定來說，足夠支援即時視訊的應用。

第二個實驗觀察在固定平均封包連續遺失長度為3的情況下，網路封包遺失率設定為2%~20%不等的受測封包遺失率以及FEC延遲時間結果。如圖7所示，ICST FEC與SCIT FEC可以達到比non-interleaving FEC還低的受測封包遺失率，隨著網路封包遺失率增加，三種FEC機制所觀察到的受測封包遺失率也會隨之增加，但ICST FEC與SCIT FEC兩者仍會保有相同的效能。如圖8所示，三種FEC機制的延遲時間大小是non-interleaving FEC最小，ICST FEC次之，SCIT FEC最大。隨著網路封包遺失率增加，因為接收端需要等待較多封包到達以進行FEC解碼，所以整體的延遲時間曲線也會稍微呈現上升的趨勢。

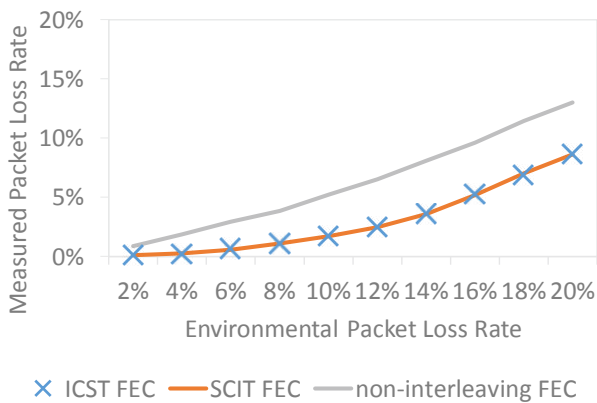


圖 7 不同網路封包遺失率下，接收端所量測到的封包遺失率

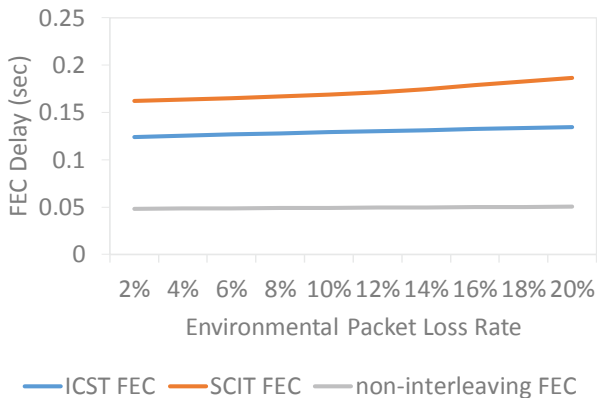


圖 8 不同網路封包遺失率下，接收端所量測到的延遲時間

圖9與圖10為前述兩個實驗的PSNR(peak signal-to-noise ratio)結果。基本上，可以驗證圖5與圖7的實驗結果，較高的受測封包遺失率會得到較低的PSNR值。從圖9與圖10中，我們仍舊可以看到ICST FEC與SCIT FEC可以得到相同的PSNR曲線；在圖9中，ICST FEC與SCIT FEC兩者跟non-interleaving FEC的差距分布於3.5 dB到1.6 dB之間，而圖10的差距則在3.9 dB到1.8 dB之間。

總結來說，有使用交錯機制的ICST FEC與SCIT FEC，從圖5、圖7以及圖9~圖10來看，在對抗連續性的網路封包遺失方面，的確可以獲得較佳資料回復的性能，但從圖6與圖8來看，

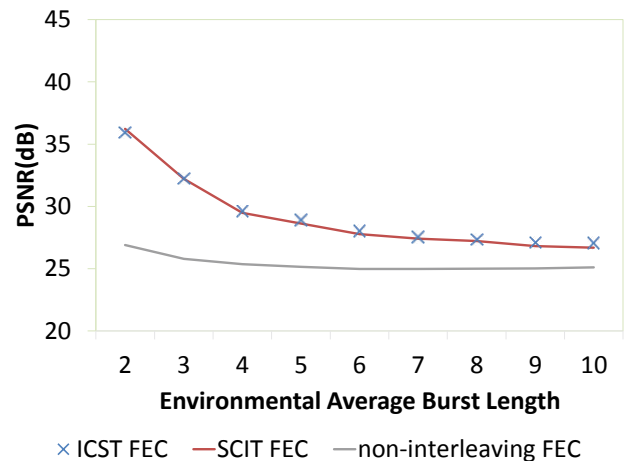


圖 9 不同連續封包遺失長度下，接收端所量測到的PSNR

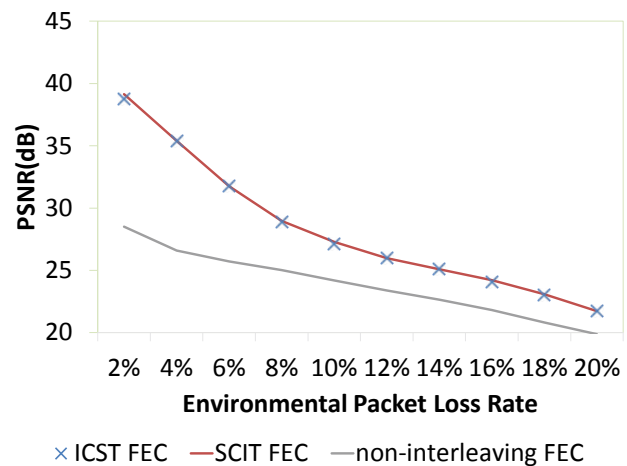


圖 10 不同網路封包遺失率下，接收端所量測到的PSNR

相對的也具備有延遲較高的缺點。本文所提出的ICST FEC在效能方面與傳統的SCIT FEC相同，並且具備較低的延遲時間。

## 5. 結論

本論文針對無線行動網路裡常遭遇到的連續資料遺失問題進行探討。傳統SCIT FEC機制使用交錯傳輸，以分散連續資料遺失的做法來增加FEC的效能，但同時也因為會大幅增加延遲時間而在應用上有所限制。本論文因此提出創新的ICST FEC機制，使用循序傳輸與交錯編碼的方法，達到低延遲性的效果。根據延遲時間分析以及模擬實驗結果，我們提出的ICST

FEC將更適合於大資料量且有即時性要求的影音串流應用。

## 參考文獻

- [1] 2013台灣數位內容產業年鑑。
- [2] (2013) Google : 臺灣與亞洲行動網路及使用者行為調查報告 [online]. Available: <http://tappier.com/google-asia-mobile-and-user-behavior-survey>
- [3] Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications Amendment 8: medium access control (MAC) quality of Service enhancements, IEEE Standard 802.11, pp.1-189, 2005.
- [4] 802.16-2004 Part 16: Air interface for fixed broadband wireless access systems, IEEE Standard 802.16, Oct 2004.
- [5] L.Rizzo, "Effective erasure codes for reliable computer communication protocols," ACM SIGCOMM Computer Comm. Rev., vol. 27, no. 2, pp. 24-36, Apr. 1997.
- [6] Chi-Huang Shih, Chun-I Kuo, Ce-Kuen Shieh, and Yeh-Kai Chou, "Small-block interleaving for low-delay cross-packet forward error correction over burst-loss channels," International Journal of Communication Systems, vol. 27, no 12, pages 3980-3995, 2014.
- [7] Tin-Yu Wu, Guizani, S., Wei-Tsong Lee, Po-Chang Huang, "An enhanced structure of layered forward error correction and interleaving for scalable video coding in wireless video delivery," IEEE Wireless Communications Magazine, vol. 20, no. 4, pp. 146-152, 2013.
- [8] Celandroni N and Gotta A. "Performance analysis of systematic upper layer FEC codes

and interleaving in land mobile satellite channels," IEEE Transactions on Vehicular Technology, vol. 60, no. 4, pp. 1887-1894, 2011

- [9] Kyungtae Kang, Cheolgi Kim, Kyung-Joon Park, "A hybrid architecture for delay analysis of interleaved FEC on mobile platforms," IEEE Transactions on Vehicular Technology, vol. 59, no. 4, pp. 2087-2092, 2010
- [10] Abdelhamid Nafaa, Liam Murphy, Tarik Taleb, "Forward error correction strategies for media streaming over wireless network," IEEE Communications Magazine, vol. 46, no. 1, pp. 72-79, 2008.

## 作者簡介

### 周業凱



國立中正大學資訊工程研究所取得工程碩士學位。現任職於工研院資通所網路視訊系統部技術副理。專長為網路多媒體串流系統與編解碼技術。

[E-mail: KennyChou@itri.org.tw](mailto:KennyChou@itri.org.tw)

### 施啟煌



國立成功大學電機工程研究所取得工程博士學位。目前擔任弘光科技大學資訊工程系專任助理教授。研究專長為多媒體網路傳輸與應用。

[E-mail: chshih@sunrise.hk.edu.tw](mailto:chshih@sunrise.hk.edu.tw)

### 鄭雯方



畢業於國立清華大學資訊工程研究所。現任職於工研院資通所網路視訊系統部。專長為視訊壓縮、網路多媒體串流系統與應用。

[E-mail: wfcheng@itri.org.tw](mailto:wfcheng@itri.org.tw)



林孟平



畢業於國立成功大學電機工程研究所。研究專長為P2P網路與多媒體傳輸。

[E-mail:jameslinjk@gmail.com](mailto:jameslinjk@gmail.com)