電腦通訊網路

Multicast Programming

Deadline:2019/04/30 23:59 上傳至Moodle

**實驗目的:**

1. 選擇在MS window或 Linux平台上，開發群播檔案傳輸伺服器與客戶端程式。
2. 加入FEC機制來做為錯誤修正碼，以提高封包接收的正確程度。

**實驗內容:**

群播是指群組內某成員一次傳送一份資料給其他成員同時接收。本實驗內容是當用戶端啟動時會加入某一個群播群組，當播放事件被觸發時(如某時間點到了，或伺服器主動傳送)，伺服器開始傳送資料給群組內所有成員，並使用FEC機制來修正傳輸過程產生的錯誤。本實驗將學習，(1)群播程式實作(2)FEC機制

1. 群播的位置使用:

群播主要目的是服務伺服器傳送單一訊息，就能讓所有的用戶端接一起接收到訊息，不必一一將訊息傳送給目的用戶端，以減少頻寬的使用。

群播需一組特殊的IP位址，此稱為多點傳播位址，在IPv4多點傳播位址是屬於Class D，範圍從224.0.0.0到239.255.255.255，下表列出目前IPv4中多點傳播保留位址。

|  |  |
| --- | --- |
| 位址 | 用途 |
| 224.0.0.0 | (保留) |
| 224.0.0.1 | 子網路上所有節點 |
| 224.0.0.2 | 子網路上所有路由器 |
| 224.0.1.1 | NTP通訊協定 |
| 224.0.0.9 | RIPv2協定群播位址 |
| 224.0..1.24 | WINS伺服器群播位址 |

1. FEC(Forward Error Correction):FEC是一種單向通訊修正技術常應用於多媒體資料傳輸，FEC是透過冗餘資訊來修正遺失的錯誤，常見的FEC演算法有漢明碼、BCH碼、ReedSolomo編碼與里德-所羅門誤碼校正編碼等等。

**實驗設備:**

1. 伺服端:一台裝有MS windows/ Linux的機器。用戶端:一台裝有MS windows/ Linux的機器或是行動裝罝。
2. 可以利用visual C++/C#或gcc或Java來開發程式 (使用Socket programming，嚴禁使用相關已包裝好的網路元件)。

**實驗情境:**

實現一個允許多人連線的群播應用伺服器程式，具以下功能 :

* 1. 允許多個Client端，同時進行連線、並加入某一群組。
  2. 由伺服端主動觸發播送事件，進行多媒體檔案資料傳輸，並紀錄封包傳輸過程中的遺失率等資訊

1. 伺服器與客戶端在**有線**環境下做資料傳輸，並記錄其遺失率。
2. 伺服器與客戶端在**有線**環境下做資料傳輸加入FEC機制，並記錄其遺失率。

**實驗成果及報告:**

第一階段:

實作一個多媒體檔案群播的應用程式，同時允許多個使用者連線，。

第二階段:

當播放事件被觸發時，多媒體檔案群播利用multicast技術，傳送多媒體資料。並在封包中加入**序號**，依本實驗所列四個實驗情境進行實驗，並利用序號來計算封包遺失率。

**實驗技術與API:**

* 1. FEC library可參考“C++ forward error correction library with SIMD optimizations”[5]，這是一個FEC的實作，實作了encoder與decoder。

**參考文件:**

1．Linux網路程式設計 黃俊翔 譯，碁峰

2．Microsoft Windows網路程式設計第二版 陳弘馨 譯，Microsoft press

3．Pthread多緒程式設計 蕭伯剛 編譯，O’Reilly

4．網路應用程式設計介面Socket與XI 林慶德 譯，培生

5．C++ forward error correction library https://github.com/randombit/fecpp

**分數分配：**

|  |  |
| --- | --- |
| Unicast transfer(code需附上註解) | 20% |
| Multicast transfer(code需附上註解) | 20% |
| Multi-media transfer(code需附上註解) | 20% |
| FEC(code需附上註解) | 20% |
| 報告書 | 20% |
| Total | 100% |