國立暨南國際大學資訊工程學系

碩士論文

命名資料網路 (NDN) 中網頁即時通訊 (WebRTC) 技術之設計與基於 PyQt5 之實作

The Design and Implementation of Web-based Real-Time Communication (WebRTC) Technology in Named-Data Networking (NDN) with PyQt5

指導教授：吳坤熹博士

研究生：王琮閔

中華民國108年9月

doi:10.6837/ncnu201900270

國立暨南國際大學碩士論文考試審定書

資訊工程 學系（研究所）
研究生 王琮閔 所提之論文

命名資料網路 (NDN) 中網頁即時通訊 (WebRTC) 技術之設計與基於 PyQt5 之實作

The Design and Implementation of Web-based Real-Time Communication (WebRTC) Technology in Named-Data Networking (NDN) with PyQt5

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

朱守仁 委員兼召集人

吳坤熹 委員

張議杰 委員

中華民國 一〇八 年 七 月 十五 日

doi:10.6837/ncnu201900270

致謝詞

首先，第一個要感謝的人是就學期間教導我的 吳坤熹 老師。剛進到研究所的第一年，在「研究生全完求生手冊」當中，指導教授的選擇對於一名研究生來說，影響的不僅僅是研究，更深的是各種未來就業所需的技能培養。我非常慶幸當時選擇了吳坤熹教授作為我的指導教授，儘管在第一年的時候讓我非常不能適應，讓我一度在想，為什麼讀研究所這麼辛苦，為什麼一起進來研究所的同學可以那麼輕鬆。

老師曾經說過，每個禮拜要我們整理的報告，不只是為了讓他了解到目前的進度與研究方向，更是讓我們學習如何將技術與經驗記錄下來，在我們需要這些知識的時候才不會顯得手足無措。這份報告同時也記錄了，我們在研究所的學習經歷。當我重新審視了這三年來，我所做的這些報告之後，我才明白，就算非常辛苦，也是促使我成長的養分。

在做研究的路上，我時常迷惘到底該做些什麼；而吳教授都會根據當時的狀況給予我所缺乏的。教授常說:「做研究就是要站在巨人的肩膀上」，除了避免閉門造車的窘境，更重要的是，吸取前人的智慧並且提出哪裡好、哪裡不好、哪裡可以改進，養成自己批判性的思考。吳教授讓我重新認識到該怎麼去完成一件事，相信這在我未來的求職路上，定能成為我的一大助力。

接著是想感謝國呈學長跟俊杰學長，在我剛進實驗室的時候, 給予我許多的幫助，不管是實驗室的事務還是在研究所的生活上，我非常感謝他們的照顧。如果沒有他們兩人的幫助，我想會花上更多時間才能完成研究。

最後我想感謝我的母親以及阿姨，在父親去世的時候把家裡撐了下來，讓我能繼續完成我的學業。期許自己在未來能好好地用自己所學來回報我的家人們。

doi:10.6837/ncnu201900270

論文名稱：命名資料網路 (NDN) 中網頁即時通訊 (WebRTC) 技術之設計與基於 PyQt5 之實作

校院系：國立暨南國際大學科技學院資訊工程學系　　頁數：77

畢業時間：中華民國 108 年 9 月　　學位別：碩士

研究生：王琮閔　　指導教授：吳坤熹博士

摘要

命名資料網路為一種未來的網路架構，以命名資料取代傳統的 IP 位址。命名資料網路有將資料快取在網路節點的機制，當節點收到要求相同內容的封包時，可以直接從節點得到資料，節省封包到服務器的時間與流量。在傳統 IP 網路中，影音串流服務需要回應每位使用者的要求，除了服務器工作量繁重之外，流量也非常龐大。其中，像是新聞、比賽等直播的串流屬於大量且要求內容相同，命名資料網路所具備的快取機制，就相當適合應用在此服務上以降低系統負擔。

為了推廣命名資料網路的運作，本篇論文提出了一個網頁即時通訊的應用，使用命名資料網路進行訊息交換及語音通訊，使用者只需要簡單的操作即可透過與他人聊天。即時通訊為日常生活中相當重要的工具，而命名資料網路在瀏覽器方面的應用仍相當缺乏，本論文的貢獻是希望藉由在命名資料網路的瀏覽器應用上，增加實用的功能，吸引更多人使用，藉此促進命名資料網路的發展 。

關鍵字：命名資料網路、NDN 瀏覽器、網頁即時通訊

doi:10.6837/ncnu201900270

Title of Thesis: The Design and Implementation of Web-based Real-Time Communication (WebRTC) Technology in Named-Data Networking (NDN) with PyQt5

Name of Institute: Department of Computer Science and Information Engineering, College of Science and Technology, National Chi Nan University

Pages: 77

Graduation Time: 09/2019 Degree: Master

Student Name: Tsung-Min Wang Advisor Name: Dr. Quincy Wu

Abstract

Named-Data Networking is a network architecture in the future. It adopts data names to replace IP addresses. In delivering data, NDN could cache data in network nodes. When a network node in NDN receives a packet which tries to retrieve data requested by someone previously, the network node can reply the data directly without further contacting the original data source. In this way, the request need not arrive at the server. This mechanism significantly reduces time and network traffic. In modern Internet, video streaming servers need to respond to each request, which causes heavy loading on servers. The mechanism of NDN is suitable to reduce the heavy loading caused by duplicate requests.

Although NDN is a good mechanism, it still lacks rich applications. This thesis proposes a Web-based Real-Time Communication application in NDN, where users can exchange text messages and voice chats easily. With this useful tool, people can use NDN easiy.

Keywords: Named-Data Network, NDN-Browser, WebRTC

doi:10.6837/ncnu201900270

目次

致謝詞 i

摘要 ii

Abstract iii

目次 iv

表目次 vii

圖目次 viii

第一章 緒論 1

第一節 研究動機 1

第二節 研究目的 8

第二章 背景知識 11

第一節 Named Data Networking (NDN) 11

第二節 NDN 網路節點運作 14

第一小節 網路節點運作與快取機制 14

第二小節 NDN Multicast 19

第三節 Web Cache 20

第四節 Multicast 22

第五節 WebRTC 25

第六節 技術挑戰 25

第三章 NDN 網頁即時通訊 27

第一節 系統架構 27

第二節 Web Producer 模組介紹 27

第一小節 Web Producer 28

第二小節 IP 與 NDN 網頁存取服務差異 29

第三小節 NDN Browser 模組介紹 31

第四小節 Web Consumer 模組介紹 32

第五小節 Web Consumer 模組運作說明 33

第三節 應用架構 34

doi:10.6837/ncnu201900270

第一小節 IP 與 NDN 差異 34
第二小節 NDN 網頁即時通訊應用 38
第四節 操作流程 39
第一小節 Alive Module 40
第二小節 Register Module 41
第三小節 Search Module 41
第四小節 Session Module 43
第五小節 Message Module 44
第六小節 VoiceChat Module 45
第四章 系統實作 48
第一節 網路架構 48
第二節 NDN Node 49
第一小節 WireShark 新增對 NDN 封包的解析 50
第二小節 TCP Tunnel 的建立 50
第三小節 WebSocket Tunnel 建立 51
第四小節 NFD 安裝 52
第五小節 允許 Tunnel 建立 52
第六小節 重啟 NFD 52
第三節 Web Producer 52
第一小節 安裝 PyNDN 53
第二小節 設定 Web Producer 53
第三小節 Web Producer 實作 54
第四節 Web Consumer 55
第一小節 安裝 PyQt5 與 PyNDN 55
第二小節 PyQt5 與 PyNDN 函式介紹 55
第三小節 設定 WebConsumer 56
第四小節 Web Consumer 實作 57
第五節 網頁即按即說應用 (Multicast) 59
第一小節 IP 網路的即按即說應用 60

doi:10.6837/ncnu201900270

第二小節 NDN 即按即說應用 62
第五章 實驗結果 67
第六章 結論及未來展望 73
參考資料 74
附錄 76
附錄一 Wireshark 安裝 76
附錄二 NDN Multicast Suppression Duration 77

doi:10.6837/ncnu201900270

表目次

表一、Content Store 15
表二、Pending Interest Table 16
表三、Forwarding Information Base 17
表四、回傳 Data 後的 CS 與 PIT 18

doi:10.6837/ncnu201900270

圖目次

圖一、 SANDVINE 全球應用類別流量分析圖 1
圖二、 向伺服器要求資料 (IP Network) 2
圖三、 伺服器回應資料 (IP Network) 2
圖四、 Proxy 伺服器運作機制 3
圖五、 CDN 運作機制 3
圖六、 Content Delivery Network 4
圖七、 向資料提供者要求資料 (NDN) 5
圖八、 資料提供者回應資料 (NDN) 5
圖九、 新的使用者要求資料 6
圖十、 設定 Proxy 伺服器 6
圖十一、 Mosaic 瀏覽器 7
圖十二、 NDN 實驗室回應 8
圖十三、 NDN Browser 測試網頁 9
圖十四、 Hourglass Architecture 11
圖十五、 要求資料 (IP 網路) 12
圖十六、 要求資料 (NDN) 13
圖十七、 NDN 網路節點運作 (要求資料) 14
圖十八、 NDN 網路節點運作 (回傳資料) 18
圖十九、 要求資料 19
圖二十、 回應資料 19
圖二十一、 WCCP 網頁快取運作 20
圖二十二、 NDN 網頁快取運作 21
圖二十三、 發送給多名使用者 23
圖二十四、 Multicast 24
圖二十五、 WebRTC 25
圖二十六、 系統架構圖 27
圖二十七、 Web Producer 運作流程 28

doi:10.6837/ncnu201900270

圖二十八、 網頁要求流程圖 (IP 網路) 29
圖二十九、 網頁要求流程圖 (NDN 網路) 30
圖三十、 NDN Browser 架構 31
圖三十一、 Web Consumer 32
圖三十二、 即時通訊信令流程 (IP 網路) 34
圖三十三、 即時通訊資料走向圖 (IP 網路) 35
圖三十四、 即時通訊信令流程 (NDN) 36
圖三十五、 即時通訊流程圖 (NDN) 37
圖三十六、 NDN 網頁即時通訊應用架構 38
圖三十七、 NDN 網頁即時通訊應用流程圖 39
圖三十八、 要求語音通訊流程圖 40
圖三十九、 Alive Module 信令流程 40
圖四十、 Search Module 信令流程 42
圖四十一、 Session Module 信令流程 (接受、拒絕) 43
圖四十二、 Session Module 信令流程 (沒有回應、已在會話中) 43
圖四十三、 Message Module 信令流程 44
圖四十四、 VoiceChat Module 信令流程 (接受、拒絕) 45
圖四十五、 VoiceChat Module 信令流程 (沒有回應) 45
圖四十六、 VoiceChat Module 信令流程 (聲音的傳送) 46
圖四十七、 NDN 網路架構 48
圖四十八、 網路架構 49
圖四十九、 Wireshark 新增 NDN 封包解析 (init.lua) 50
圖五十、 建立 Tunnel (TCP) 51
圖五十一、 建立 Tunnel (WebSocket) 51
圖五十二、 取消 localhop_security 註解 52
圖五十三、 設置 NDN Node 的 IP 位址 (webProducer.conf) 53
圖五十四、 設置 NDN Node 的 IP 位址 (www/js/main.js) 54
圖五十五、 執行 Web Producer 54
圖五十六、 設置 NDN Node 的 IP 位址 (webConsumer.conf) 56

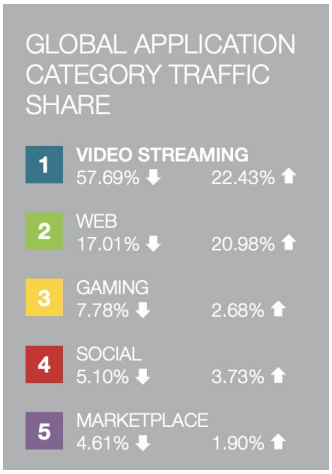
doi:10.6837/ncnu201900270

圖五十七、 執行 Web Consumer 57
圖五十八、 NDN 網頁即時通訊應用畫面 57
圖五十九、 即按即說應用 60
圖六十、 即按即說應用操作流程 61
圖六十一、 NDN 網頁即按即說應用 62
圖六十二、 群組使用者名單 63
圖六十三、 NDN 網頁即按即說應用畫面 63
圖六十四、 NDN 要求發話權 64
圖六十五、 NDN 要求語音訊息 65
圖六十六、 NDN 回應語音訊息 65
圖六十七、 NDN 釋出發話權 66
圖六十八、 TCP 連線建立 67
圖六十九、 第一次要求網頁 68
圖七十、 第二次要求網頁 68
圖七十一、 NDN 網頁即時通訊應用操作 69
圖七十二、 WebSocket 連線建立 70
圖七十三、 Alice 按下發話按鈕 70
圖七十四、 使用 Display Filter 70
圖七十五、 要求語音訊息 71
圖七十六、 要求語音訊息 (過濾後) 71
附圖一、允許 non-superuser 使用 76

doi:10.6837/ncnu201900270

第一章 緒論

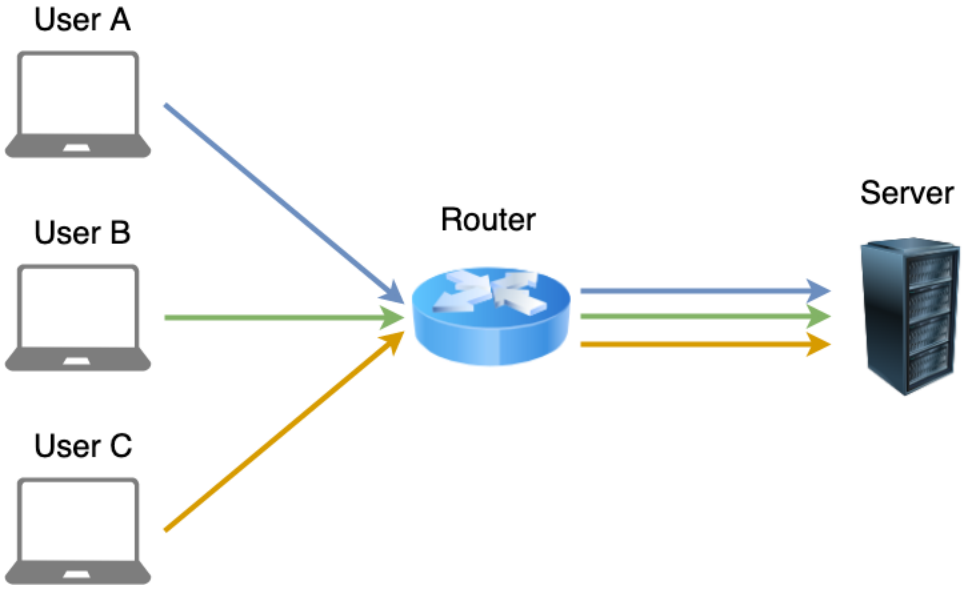
第一節 研究動機



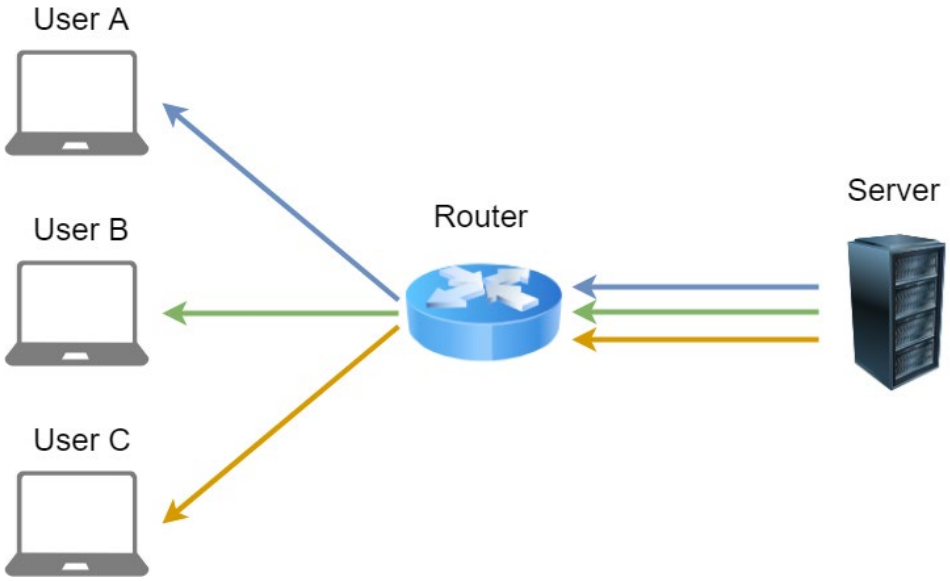
圖一、SANDVINE 全球應用類別流量分析圖

近年來，由於個人電腦及智慧手機的普及，網路已經成為日常生活的一部分。使用網路能做到的事情不勝枚舉：電子郵件、搜尋引擎、網路購物等，不管哪一項都帶來了莫大的便利性。影音服務更是讓使用者可以在任何時間、任何地點，用自己的設備聽音樂或是看影片。其中，YouTube、Netflix 更是世界知名的影音服務平台，不管是短片、直播、連戲劇還是電影，都可以在平台上面觀賞。SANDVINE 於 2018 年提出的 The Global Internet Phenomena Report，其中根據應用類別所分析出的流量占比（圖一），光是 Video Streaming 的下載流量就高達 57.69 %。

doi:10.6837/ncnu201900270



圖二、向伺服器要求資料 (IP Network)



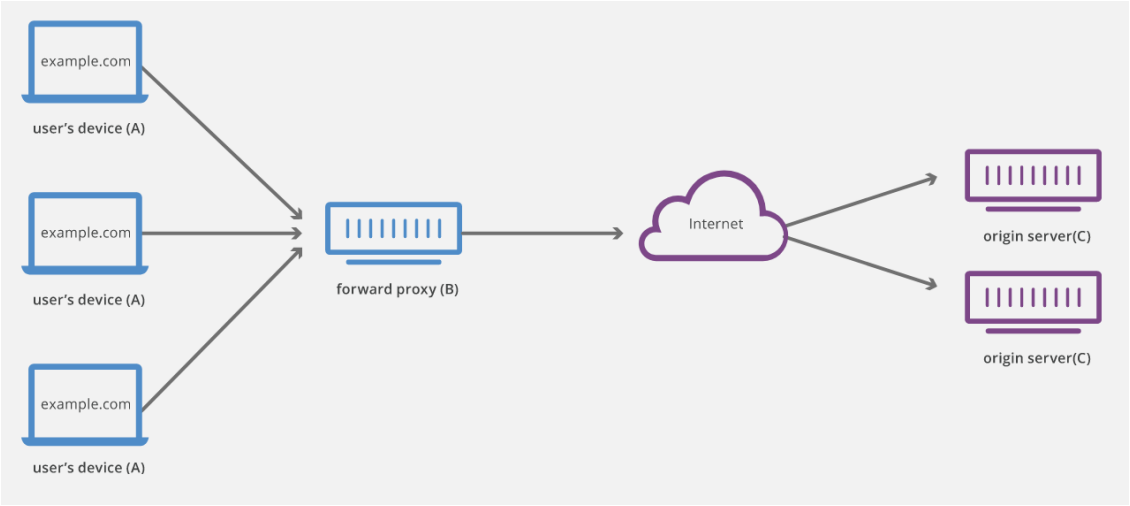
圖三、伺服器回應資料 (IP Network)

在傳統的網路上，假設在 YouTube 上的一部影片，有三位使用者想看，此時，YouTube 的伺服器就會收到三份相同的要求，如圖二所示。

接著，伺服器將影片分別發送到這三人的設備上。圖三可以看到，儘管三位使用者要求的是同一部影片，伺服器依舊需要回應三次相同的內容給使用者。如果這些封包都是相同的內容，是不是可以讓伺服器只回應一次就好？ 在 IP 網

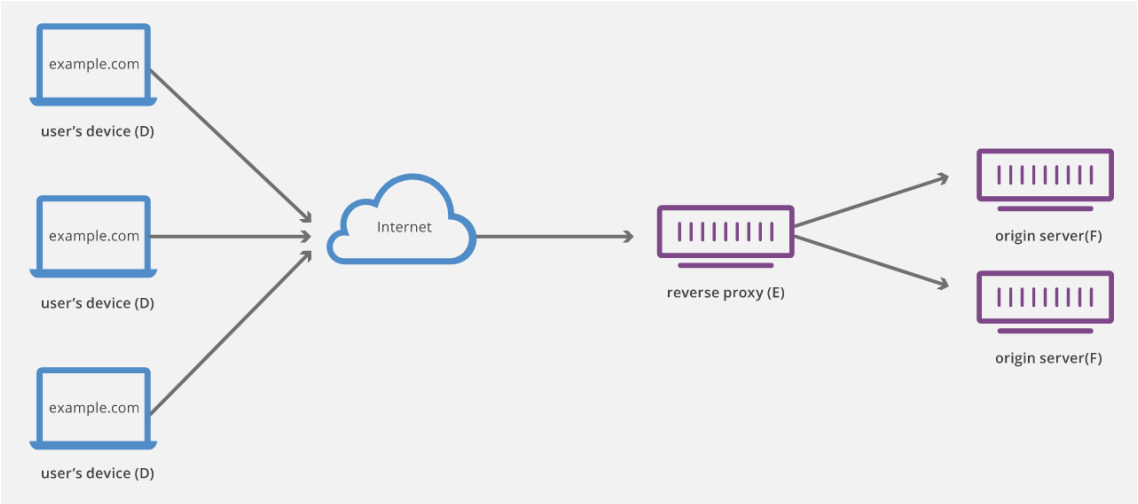
doi:10.6837/ncnu201900270

路中，過去是使用 Proxy 伺服器或 Content Delivery Network (CDN) 來解決這樣的問題。



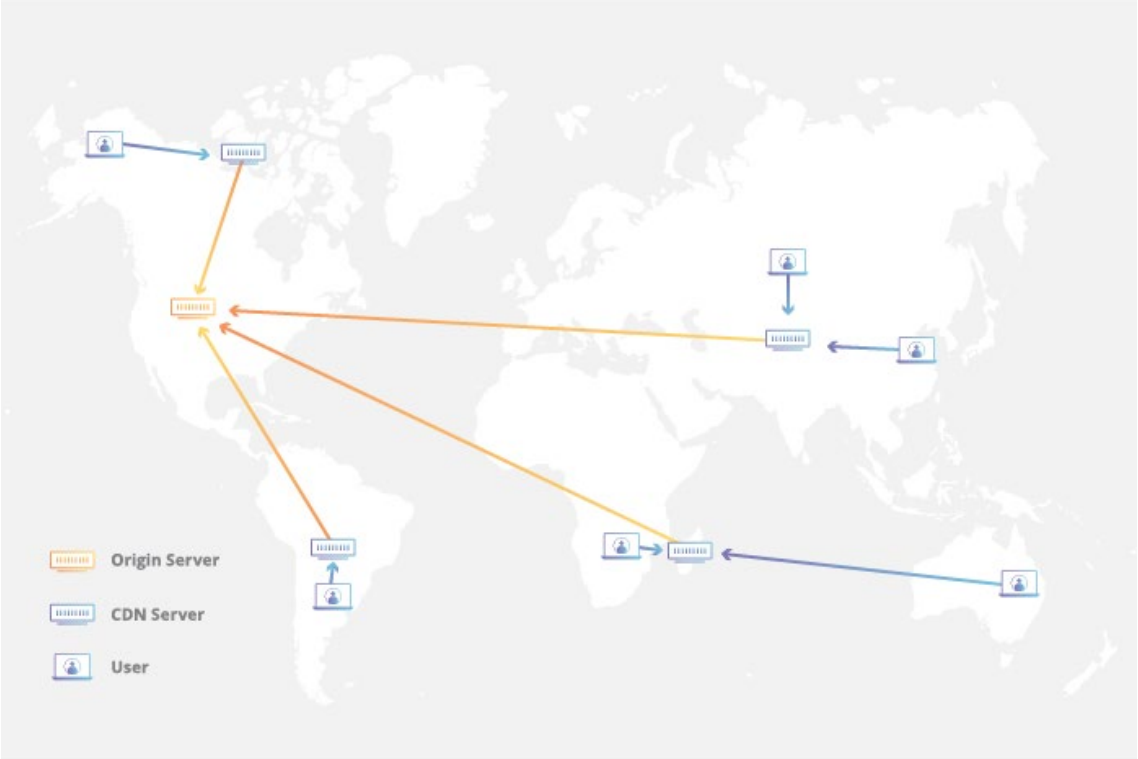
圖四、Proxy 伺服器運作機制

[1] 提到了 Proxy 伺服器與 CDN 的運作方式，圖四為 Proxy 伺服器的運作示意圖。從圖上可以看到，多位使用者 (user`s device) 會透過一台 Proxy 伺服器 (forward proxy) 向提供資料的伺服器 (origin server) 發送要求。由於提供資料的伺服器只須回應 Proxy 伺服器一次，因此可以達到節省流量的效果。此外，Proxy 伺服器還會將已經要求過的資料快取一份下來，當有新的使用者要求同一份資料的時候，便能直接從 Proxy 伺服器上取得，無需再度詢問資料伺服器，有效減輕資料伺服器的負擔。



圖五、CDN 運作機制

doi:10.6837/ncnu201900270

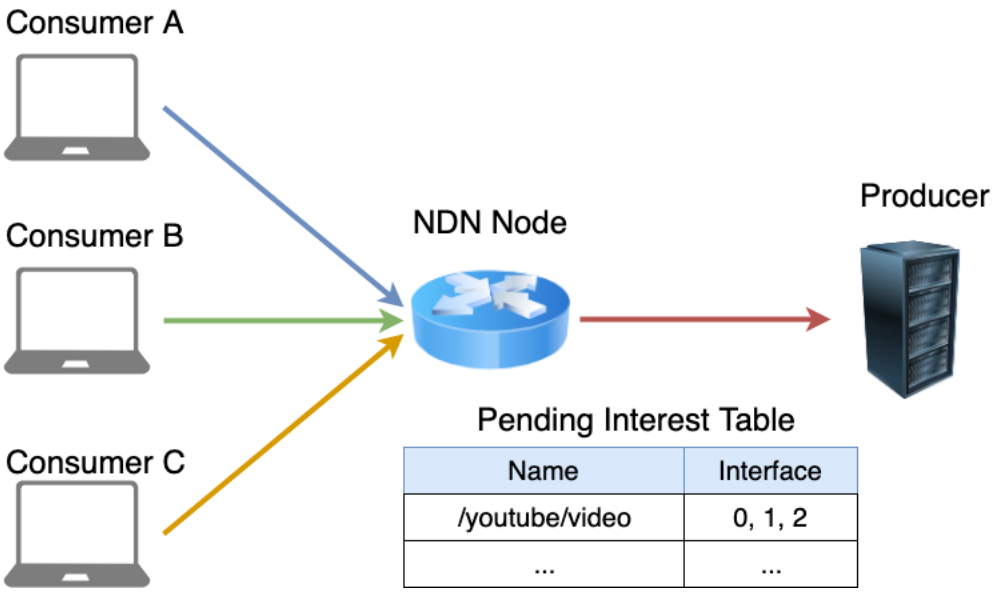


圖六、Content Delivery Network

[2] 詳細地說明 CDN 的功能，其屬於反向的 Proxy 伺服器 (reverse proxy)，服務方會讓 CDN 的 Proxy 伺服器先將資料快取下來，當使用者 (user`s device) 要向提供資料的伺服器 (origin server) 進行要求時，會由 CDN 中的 Proxy 伺服器代為回應，如圖五所示。

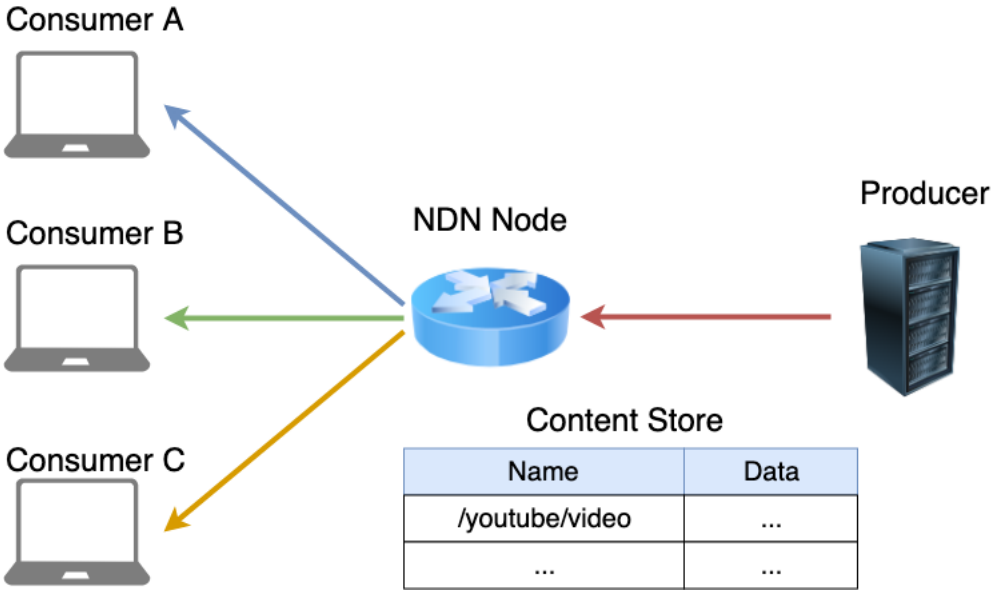
CDN 透過在各地部署 Proxy 伺服器並且預先將資料進行快取，讓使用者可以直接從最近的伺服器上取得資料，藉此讓使用者獲得更快的反應速度，如圖六所示。另外，由於 CDN 會代替回應，也能減少服務方伺服器的負擔。

doi:10.6837/ncnu201900270



圖七、向資料提供者要求資料 (NDN)

NDN 作為未來網路架構的其中之一，其網路節點具備了快取資料的功能，讓已經要求過的資料內容，可以直接從網路節點取得。並且能將相同的要求記錄下來，僅送出一份要求到 Producer。而這樣的機制可以帶來什麼樣的好處呢？接下來同樣以三位使用者要求同一部影片的例子來觀察。

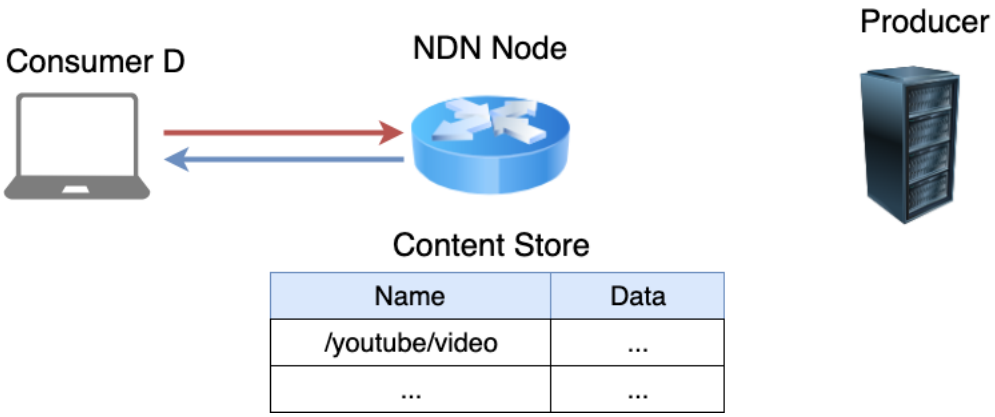


圖八、資料提供者回應資料 (NDN)

使用者 (Consumer) 發出要求 (Interest) 後，NDN 節點只會轉送第一份要求到資料提供者 (Producer)，其餘相同要求的介面會記錄在節點上的 Pending Interest

doi:10.6837/ncnu201900270

Table (PIT)，如圖七所示。接著，當節點收到資料提供者的回應後，會根據其記錄 (PIT) 上的資訊，把回應 (Data) 送到每一位使用者的手上，並且快取一份在 Content Store (CS)，如圖八所示。



圖九、新的使用者要求資料

若是有新的使用者要求同一份影片的時候，由於 NDN 節點先前已經將資料內容快取下來了，因此，使用者可以直接從節點上取得，如圖九所示。從上述的例子可以看到，NDN 在網路層上有著相當不錯的機制以改進其效能。

至於 IP 網路之中的 Proxy 伺服器，同樣也是代替使用者去向伺服器要求資料，並且快取一份在 Proxy 伺服器上。

圖十、設定 Proxy 伺服器

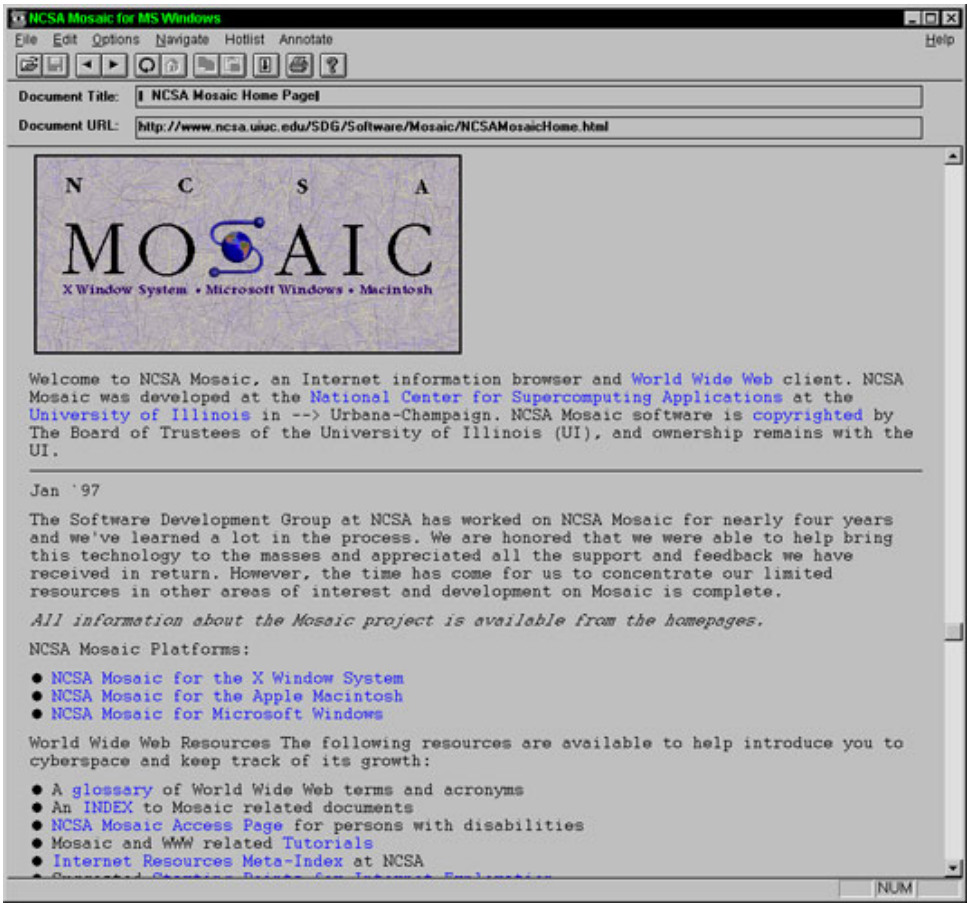
doi:10.6837/ncnu201900270

但 Proxy 伺服器需要架設，且使用者必須主動在電腦上指定使用，如圖十所示。再加上，若是使用者從不同的地方上網，指定的如果仍是同一台 Proxy 伺服器，就有可能造成兩者之間的距離過遠，導致效果不佳。

NDN 的每一個網路節點上都具備了快取的功能，就算使用者由不同的地方上網，NDN 都可以判斷哪一個網路節點具備使用者所要求的資料，讓使用者可以從最近的網路節點上取得資料內容。

另外，CDN 是透過在各地部屬 Proxy 伺服器並且預先快取資料，讓使用者能從更近的伺服器上取得資料。NDN 則是所有的網路節點都能快取並且回應，若是該資料已被要求過，使用者亦可從最近的網路節點上取得資料，達到與 CDN 相同的效果。

儘管 NDN 的架構有著為整個網路世界帶來改變的可能，但沒有人使用也只不過是眾多「未來」網路架構的其中之一。那 NDN 要如何呈現，才有辦法去吸引到世人的眼光呢？



圖十一、Mosaic 瀏覽器

doi:10.6837/ncnu201900270

根據 1993 年，Marc Lowell Andreessen 所釋出的 Mosaic 瀏覽器 [3]（圖十一），一出世就讓眾人為之瘋狂。其簡易的操作再加上網際網路那宛若沒有盡頭的龐大資訊量，人們無不前仆後繼地進入網路的世界。同時，這也使得網路整體開始蓬勃的發展。時至今日，網路已成為我們生活中不可或缺的一部分。本篇論文認為，這正是 NDN 所需要借鏡的例子。

有鑑於此，在 NDN 之中，[4] 所提出的 NDN Browser，雖然已具備在 NDN 瀏覽網頁的功能並且提供使用者容易操作的介面。但相較於現今 IP 網路上的瀏覽器已經是非常的成熟，不管什麼樣的功能都可以在瀏覽器上看見。還在成長階段的 NDN 瀏覽器，功能與便利性上都還差上一大截。但，NDN 可以從 IP 網路上學到，用什麼樣的工具與表現方式是容易被大眾所接受的。

第二節 研究目的

本研究之目的是在 NDN Browser 的基礎上，提出一套網頁即時通訊應用，並且使用 NFD (Named-Data Networking Forwarding Daemon)，負責提供 NDN 的網路功能。

Hi Ken

It's sad that NDN doesn't support one of the most popular apps - web.
We used to have a Firefox add-on
<https://github.com/named-data/ndn-js/tree/cf535396cb4242dd8ebfb020ca622481eb56466b#firefox-add-on-for-the-ndn-protocol>
but
I haven't been able to install it in recent Firefox versions.

I have some thoughts on how to deploy NDN websites:
https://yoursunny.com/t/2011/ndn-web/
Caution: it's an old article I wrote during my first month reading about NDN, so that some terminology could be inaccurate.

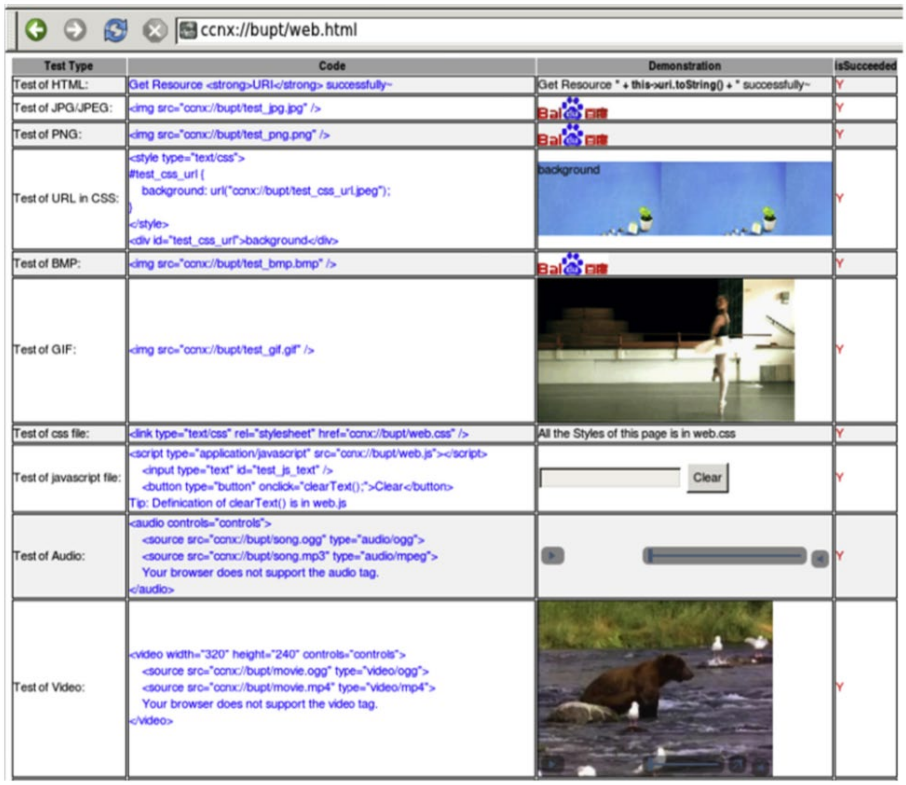
Yours, Junxiao

圖十二、NDN 實驗室回應

根據 UCLA NDN 實驗室的回應，如圖十二所示 [5]，當前 NDN 上的 Web 服務十分缺乏。 雖然先前有利用在 Firefox 上添加外掛程式 (add-on) [6] 來完成

doi:10.6837/ncnu201900270

NDN 傳輸的 Web 服務。但，隨著 Firefox 的更新，該工具在 Firefox 最近的幾個版本上都無法成功執行。



圖十三、NDN Browser 測試網頁

圖十二為 NDN Browser 的網頁顯示與支援的標籤測試。其網路環境為 CCNx [7]，是由 PARC [8] 所提供的。在這張圖上可以看到，NDN Browser 能輸入 ccnx:// 協定的 URL，且在常見的標籤亦可使用 ccnx:// 作為來源。透過這兩者的實現，讓 NDN Browser 能以 CCNx 作為底層的傳輸，完成對網頁的要求與建構。

但是，NDN Browser 使用的網路環境為 CCNx，而非 NDN 目前所提供的 NFD [9] (NDN 從 CCNx 的其中一個分支上，開發出另一個符合 NDN 需求的網路環境)。NDN 運行在 NFD 上，用於 Web 服務的 Firefox 外掛程式 (add-on)，該工具經證實，在近幾個 Firefox 版本上皆無法成功實作。

因此，本研究參考 NDN Browser 所提出的瀏覽器架構，並且以 NFD 作為網路環境，實作出以 ndn:// 要求網頁的 Web Consumer 以及回應網頁要求的 Web Producer。此外，本研究亦在此基礎上，進一步開發出一個 NDN 網頁即時通訊的應用。

doi:10.6837/ncnu201900270

NDN 使用者可以透過 Web Consumer 向 Web Producer 取得網頁即時通訊應用。進入應用後，完成簡單的註冊，便能與其他使用者進行文字訊息交換以及語音通訊的功能，達到網頁即時通訊之目的。本研究達成功能如下：

Web Producer

1. 回應網頁要求

Web Consumer

1. 要求網頁
2. 在瀏覽器的網址列 輸入 URL 時，協定部份除原有的 ftp, http, https, 亦可接受 ndn
3. HTML 文件中 的標籤來源也可以指定使用 NDN 協定的 URL

網頁即時通訊應用

1. 檢查 ID 是否已被使用
2. 註冊使用者 ID
3. 搜尋已註冊的使用者
4. 會話的建立
5. 文字訊息的交換
6. 語音通訊

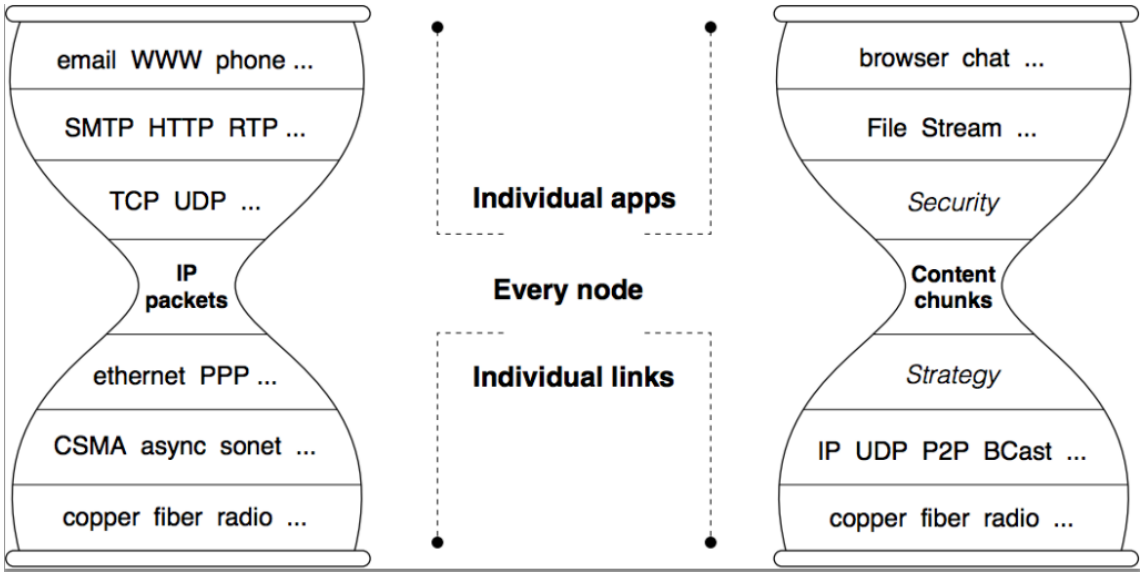
即時通訊為 IP 網路中，使用非常頻繁的功能，如 Skype、LINE、Facebook Messenger 等產品。因此，本研究在現有的 NDN Browser 基礎上，提供網頁即時通訊的應用，讓接觸到 NDN 的人，也能透過瀏覽網頁的方式，取得傳統 IP 網路中相當實用的工具。進而吸引到更多人的目光，達到推廣 NDN 的效果。

此外，NDN 在瀏覽器的應用層面非常匱乏，而本研究所提出的網頁即時通訊應用希望能幫助 NDN 補足這方面不足之處。

doi:10.6837/ncnu201900270

第二章 背景知識

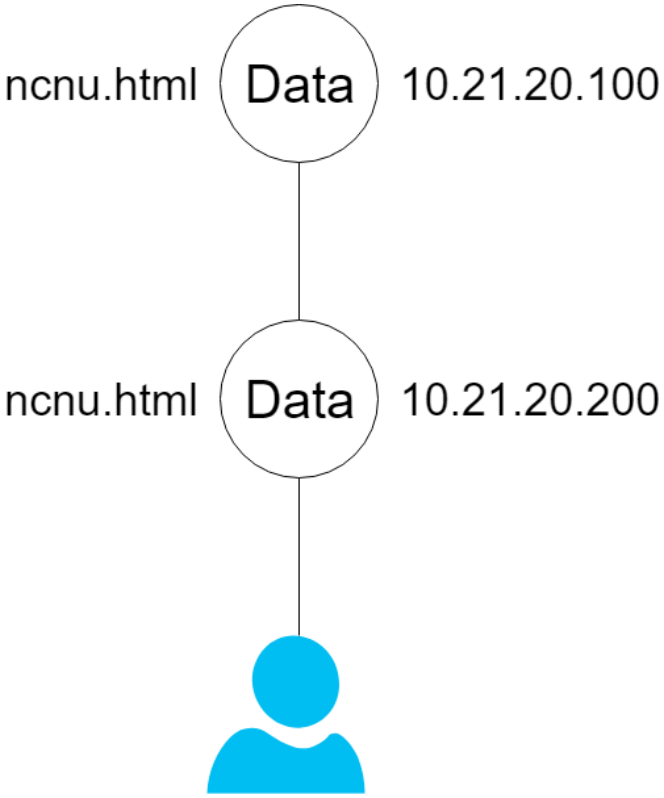
第一節 Named Data Networking (NDN)



圖十四、Hourglass Architecture

圖十四所示為現今網路的沙漏架構 (Hourglass Architecture)，是透過中間一個通用的網路層，讓各式各樣的應用能夠建立在這穩固的基礎之上，蓬勃發展。NDN 則是希望在此沙漏架構中，將瘦腰 (thin waist) ，也就是 IP 的部份，改為使用命名資料（named-data）。而這能為網路帶來什麼樣的好處呢？

doi:10.6837/ncnu201900270

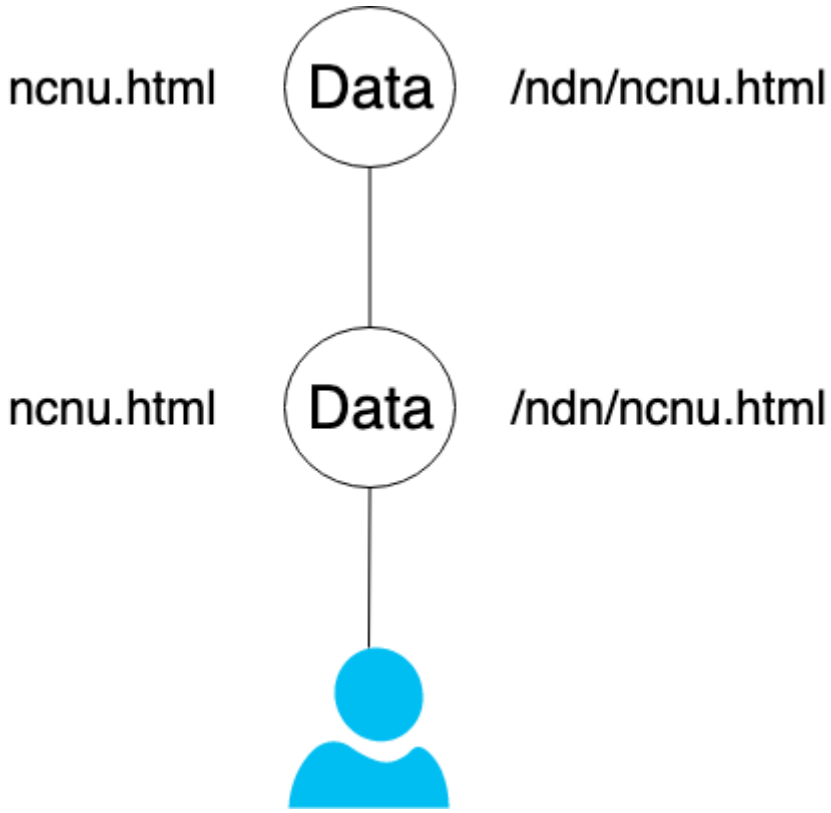


圖十五、要求資料 (IP 網路)

IP 網路中，要求資料的方式必須先選擇特定的主機，接著向其發送要求。如圖十五所示，使用者想要 ncnu.html 這份資料，此時，使用者必須選擇向 10.21.20.100 或 10.21.20.200 要求 ncnu.html。但是，明明兩台主機都有提供 ncnu.html，使用者的重點卻不是在 ncnu.html 上，而是該從哪台主機上取得。一個不小心，反而可能捨近求遠。

因此，NDN 希望透過命名資料的方式，讓資料存取的重點從特定主機上，轉移到資料內容上。

doi:10.6837/ncnu201900270



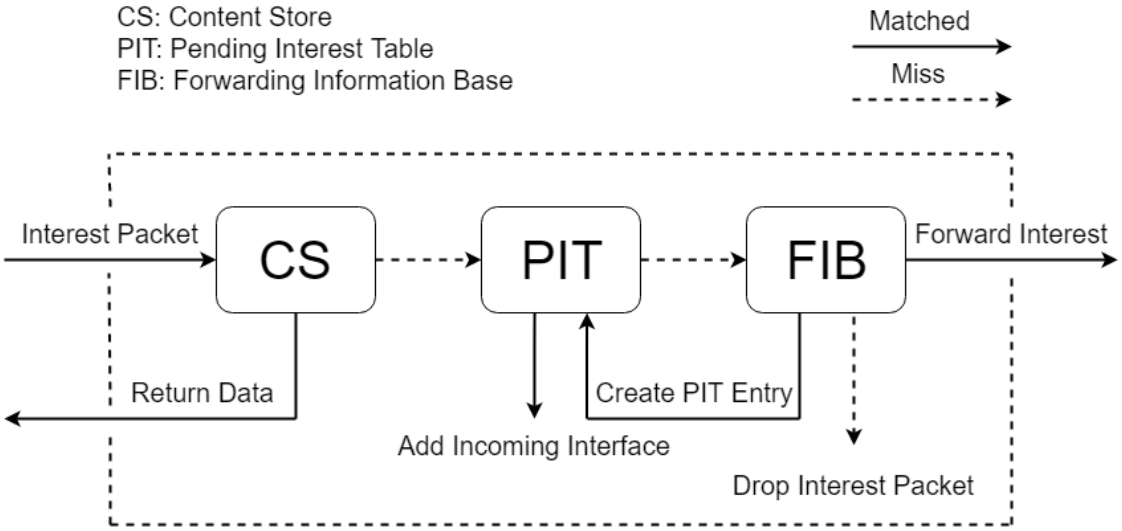
圖十六、要求資料 (NDN)

圖十六為 NDN 要求資料的方式。在 NDN 當中，提供者會在 NDN 註冊資料的名稱(/ndn/ncnu.html)，使用者只要指定該名稱，NDN 的網路節點就能根據路由資料，找到該資料的提供者，並且將資料內容回傳給使用者。NDN 將沙漏架構中的 IP 改為命名資料後，當使用者在要求資料時，就不是選擇某台特定的主機了，而是由 NDN 的網路節點自動去尋找該從哪裡取得，讓使用者能夠專注在其想要的內容上。在下一個章節中，會介紹到 NDN 的網路節點是如何達到這樣的效果。

doi:10.6837/ncnu201900270

第二節 NDN 網路節點運作

第一小節 網路節點運作與快取機制



圖十七、NDN 網路節點運作 (要求資料)

NDN 與 IP 網路在路由決定上有著很大不一樣了，IP 網路是以 IP 位址作為路由查找的基準，而 NDN 打算透過命名資料的方式取代 IP 位址。那麼 NDN 又是如何運作的呢？ NDN 裡可以分成兩種角色，分別是 Consumer 與 Producer。

Consumer 為要求的一方，以 Interest 向 Producer 進行要求，Interest 包含 Name，也就是想要要求的資料名稱。例如：用於要求本研究 NDN 網頁即時通訊應用的 Name 為 /ncnu/chat/index.html/0 (檔案會被切成多個小部分傳輸，0 為序列號碼，用於重組檔案)。

Producer 為回應的一方，以 Data 封包回應 Consumer 的要求，Producer 會在 NDN 中註冊 Prefix，只要 Interest 的 Name 有匹配到 Prefix，NDN 網路節點就會將 Interest 送至註冊該 Prefix 的 Producer。

Data 封包會包含 Name 與 Content；Name 與收到的 Interest 相同，用於表示回應該 Interest；Content 則會放入 Interest 要求的資料內容。例如：回應本研究 NDN 網頁即時通訊應用的 Data，其 Name 就會是 /ncnu/chat/index.html/0 以及 /ncnu/chat/index.html/1，而 Content 則是網頁資料的內容。

doi:10.6837/ncnu201900270

NDN 的網路節點具備三種模組：

1. Content Store (CS):

   查找 Interest 所要求的資料名稱是否已經被快取在網路節點上。

   若有，則會直接回傳 Data 給 Consumer；

   若沒有，則會交給 PIT。

2. Pending Interest Table (PIT):

   查找 Interest 所要求的資料名稱是否已經有其他人要求過。

   若有，則會在 PIT 中，同一筆要求下，新增此 Interest 進入的介面；

   若沒有，則會交給 FIB。

3. Forwarding Information Base (FIB):

   查找 Interest 所要求的資料名稱要送往哪一個介面。

   若 FIB 有 Interest 的路由資訊，則會在 PIT 新增一筆要求，接著送出；若 FIB 沒有 Interest 的路由資訊，則會將 Interest 丟棄，並且回傳 NACK 封包給 Consumer。

表一、Content Store

Name	Data
/ncnu/chat/index.html/0	...
...	...
...	...

圖十七為要求資料的運作流程。Consumer 向 Producer 進行要求的時候，會送出 Interest 封包。當 Interest 封包到達網路節點的時候，網路節點會先查找 Content Store，此 Interest 所要求的資料名稱 (Name) 是否有快取過，如表一所示。若是有的話，則會直接回傳 Data 封包給 Consumer；若是沒有的話，則會交給 Pending Interest Table。

例如：網路節點收到 Web Consumer 送出的 Interest，Name 為

doi:10.6837/ncnu201900270

/ncnu/chat/index.html/0，經查找會發現此 Name 已經被快取在 Content Store 了，因此，網路節點就會直接回傳 Data 給 Web Consumer。

表二、Pending Interest Table

Name	Interface
/ncnu/chat/index.html/1	0, 1, 2
...	...
...	...

Pending Interest Table (PIT) 收到 Interest 之後，會去查找是否已經有人要求過此資料名稱 (Name)，但是還沒收到 Data 封包，如表二所示。若是有的話，則會在相同的資料名稱下，新增此 Interest 進入的介面 (Interface)，並且等待 Data 的回傳；若是沒有的話，則會交給 FIB。

例如：網路節點收到 Web Consumer 送出的 Interest，其 Name 為 /ncnu/chat/index.html/1，並未被快取在 Content Store。網路節點會在 PIT 內查找是否有相同的要求，但尚未收到 Data。此時會查到 PIT 已記錄 /ncnu/chat/index.html/1，要求者的 Interface 為 0 和 1。因此，網路節點會將此 Web Consumer 的 Interface 添加在該筆記錄下，將其更新為 Interface 0、1、2，並繼續等待 Data 的回傳。

此外，若是 Pending 在網路節點的 Interest，其送出時間若超過 Interest 的 Lifetime，卻還沒收到 Data 的話，則會將其從 PIT 裡丟棄。而 Consumer 會在 Interest 送出時間超過其 Lifetime 後，直接判斷此 Interest 已經 Timeout。

doi:10.6837/ncnu201900270

表三、Forwarding Information Base

Prefix	Interface
/ncnu/chat	3
...	...
...	...

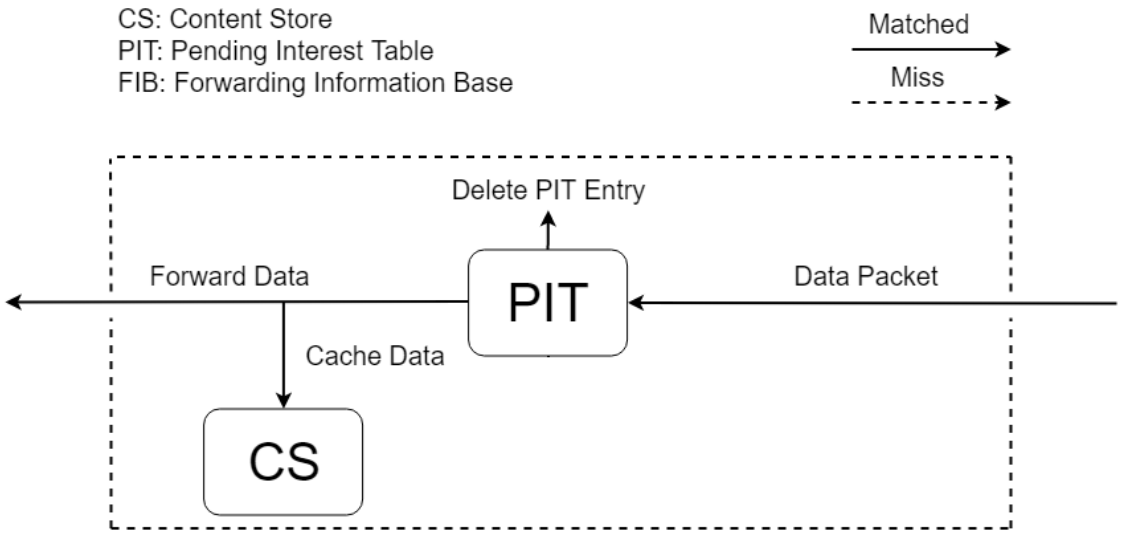
Forwarding Information Base (FIB) 收到 Interest 之後，會去查找該 Name 的路由資料，如表三所示。若是有的話，則會在 PIT 新增該 Interest 要求的資料名稱 (Name)，並且根據路由資料將 Interest 從查詢到的介面 (Interface) 送出；若是沒有的話，則會將該 Interest 丟棄，並且回傳 NACK 封包給 Consumer，告知其要求的資料內容不存在。

例如：Web Producer 註冊 Prefix /ncnu/chat，FIB 就會新增一筆路由資料 /ncnu/chat。而 Web Consumer 送出的 Interest，其 Name 為 /ncnu/chat/index.html/0、/ncnu/chat/index.html/1，會匹配 Web Producer 註冊的 Prefix /ncnu/chat。因此，網路節點會將 Interest 送至 Web Producer。

另外，當 Web Producer 註冊 Prefix /ncnu/chat，NDN 網頁即時通訊應用註冊 Prefix /ncnu/chat/User/alive 時。FIB 上就會新增 /ncnu/chat 以及 /ncnu/chat/User/alive，兩筆路由資料。若 NDN 網路節點收到 Interest，其 Name 為 /ncnu/chat/User/alive，與這兩筆路由資料都匹配時，網路節點就會判斷 Interest 的 Name 與哪一個 Prefix 匹配結果較長（Longest Prefix Match）。

也就是說 Web Producer 的 Prefix /ncnu/chat 與 Name /ncnu/chat/Alice/alive 僅匹配到 /ncnu/chat。而 NDN 網頁即時通訊應用的 Prefix /ncnu/chat/Alice/alive 與 Name /ncnu/chat/Alice/alive 匹配到 /ncnu/chat/Alice/alice。網路節點會判斷此 Interest 的 Name 與 NDN 網頁即時通訊應用的 Prefix 匹配結果較長，因此，將 Interest 送至 NDN 網頁即時通訊應用。

doi:10.6837/ncnu201900270



圖十八、NDN 網路節點運作 (回傳資料)

圖十八為回傳資料的運作流程。Producer 的 Data 封包會沿著 Interest 來時的路徑回傳。當 Data 封包到達網路節點的時候， 會由 PIT 查找，找出當初要求此 Data 的 Interest，其來自哪個介面，並且將 Data 從此介面回傳，即可達到根據 Interest 來時路徑回傳的效果。接著再由 CS 將此 Data 快取一份下來，並將此筆記錄由 PIT 刪除。

表四、回傳 Data 後的 CS 與 PIT

Content Store

Name	Data
/ncnu/chat/index.html/0	...
...	...
...	...

Pending Interest Table

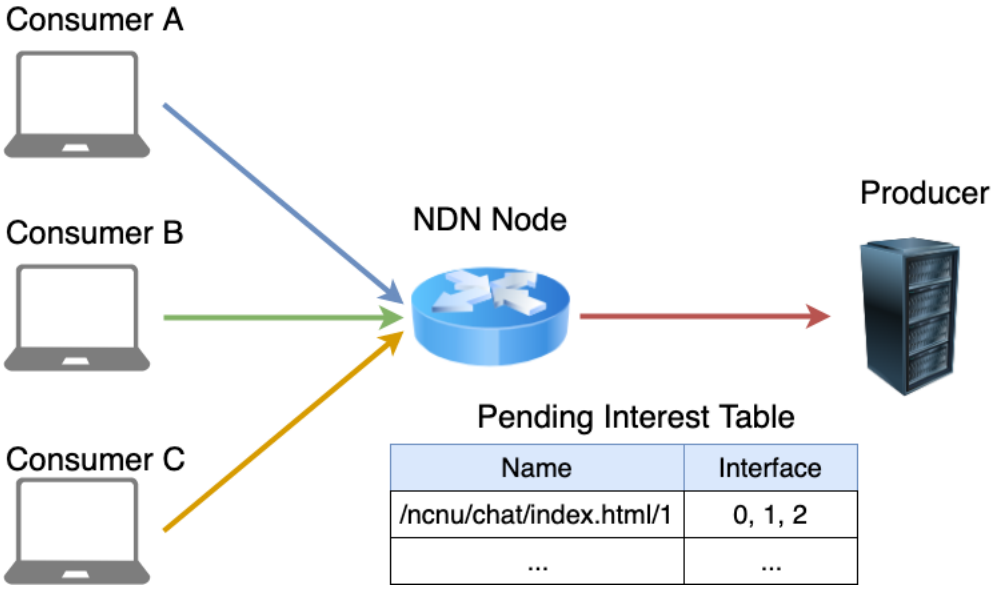
Name	Interface
/ncnu/chat/index.html/1	0
...	...
...	...

例如: Web Consumer 送出 Interest /ncnu/chat/index.html/0 以及 /ncnu/chat/index.html/1，其中，Web Producer 回應了 Data /ncnu/chat/index.html/0。此時，Pending Interest Table 就會將 /ncnu/chat/index.html/0 刪除，接著 Content Store 會快取 /ncnu/chat/index.html/0 的資料內容，如表四所示。

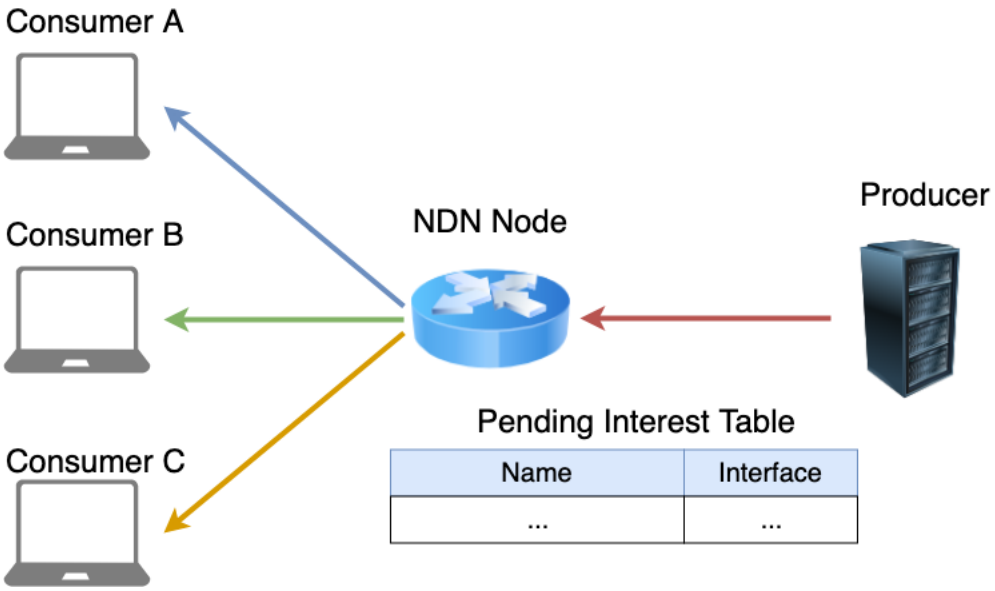
doi:10.6837/ncnu201900270

第二小節 NDN Multicast

在上一小節有提到，若是已有人要求過，但是 Data 尚未回應，則會將多筆進入介面記錄在同一要求之下。此時，當 Data 回應之後，網路節點便會將 Data 複製多份，並且向該要求下的所有介面送出 Data。完成之後，Pending Interest Table (PIT) 便會將此筆記錄刪除。這機制可以讓 NDN 具備了 Multicast 的功能。這本研究中，我們便利用一個多方即時會談的應用，來展示 NDN 的這個特性。



圖十九、要求資料



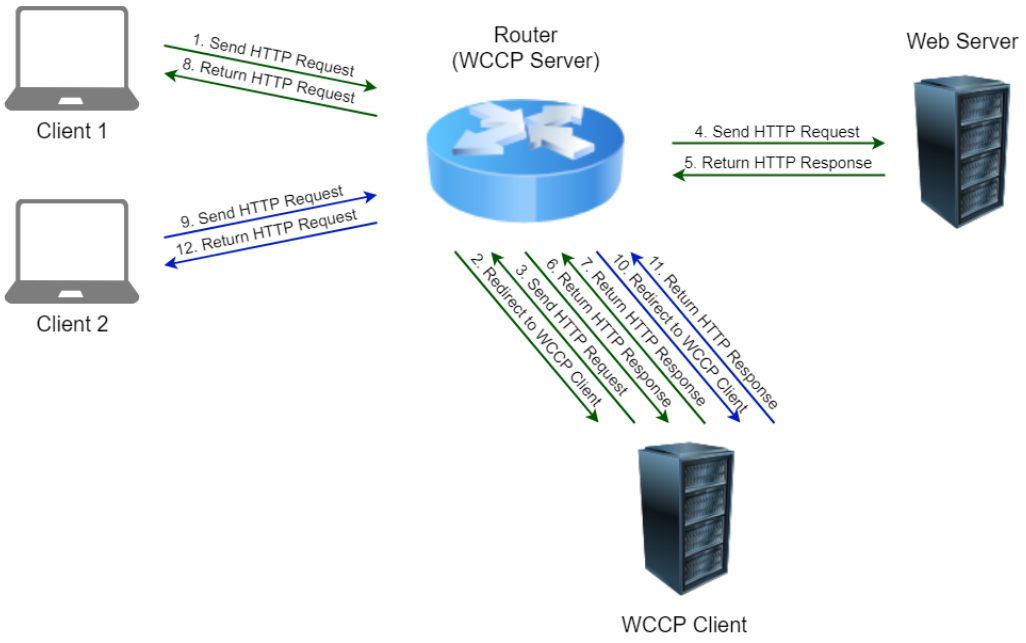
圖二十、回應資料

doi:10.6837/ncnu201900270

如圖十九所示。當三名 Consumer 要求同一份資料的時候，假設 Consumer A 的 Interest 先到達了 NDN Node，該 Interest 所進入的介面會被記錄在 PIT 內，並根據 Forwarding Information Table (FIB) 的路由資料送到 Producer。接著，當後續兩名 Consumer 的 Interest 也到達 NDN Node 時，由於 PIT 內已存在相同的要求，因此，這兩份 Interest 並不會被送到 Producer，而是將其進入的介面新增到 PIT 內的該要求之下。

如圖二十所示。當 NDN Node 收到 Producer 回應的 Data 之後，會根據 PIT 的記錄，將此 Data 複製成三份，並且回應給這三名 Consumer。NDN Node 在這裡展現出將資料同時傳遞給多名使用者，達到 Multicast 的效果。

第三節 Web Cache

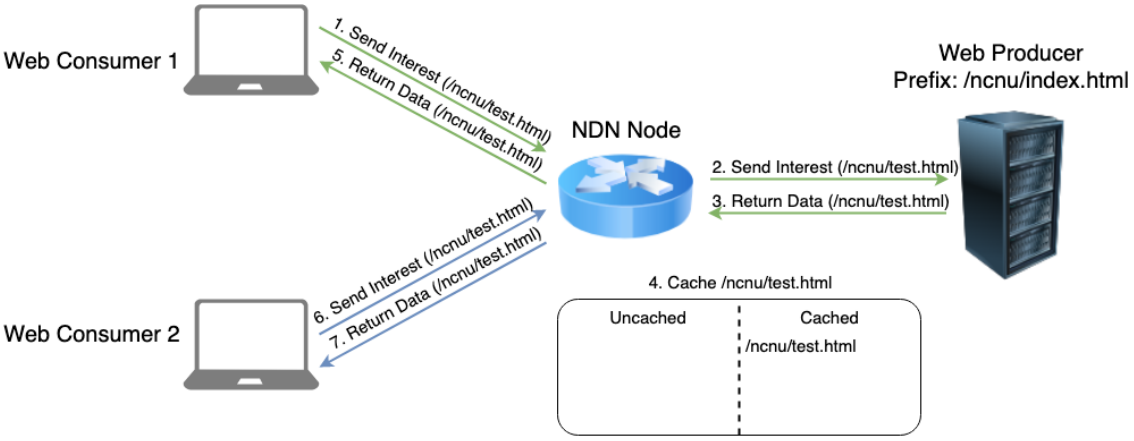


圖二十一、WCCP 網頁快取運作

IP 網路中，Cisco 的 Web Cache Communication Protocol (WCCP) [10] 為 Web Cache 協定的其中之一，是一種讓路由器具備快取功能的協定，透過將路由器與快取伺服器的結合，達到快取的效果。WCCP 可分為 WCCP Server 與 WCCP Client，WCCP Server 為一台路由器，在該路由器上設置 WCCP，則可以將網頁要求重導向至 WCCP Client，WCCP Client 為具有快取功能的機器。藉由 WCCP 的

doi:10.6837/ncnu201900270

機制，使用者就不用特地去設置一台 Proxy 伺服器，而是在路由器上就具備快取的功能。其運作流程如圖二十一所示，使用者 (Client) 的要求送到 Router 的時候，WCCP Server 會將其重導向至 WCCP Client，接著由 WCCP Client 向 Web Server 發送要求。收到 Web Server 的回傳之後，WCCP Client 便會快取一份下來，接著回傳給使用者。



圖二十二、NDN 網頁快取運作

同樣是在路由上將資料快取，NDN 的做法如圖二十二所示。使用者 (Web Consumer) 發送 Interest 封包到網路節點上，在該網路節點尚未快取過資料的情況下，會將 Interest 封包進入的介面記錄在 Pending Interest Table 上，再根據 Forwarding Information Base 的路由資訊，把 Interest 送到資料提供者 (Web Producer) 的手上。接著，網路節點會收到資料提供者所回傳的 Data。將 Data 回傳給使用者之前，網路節點會將 Data 記錄在 Content Store 裡。當有另外一名使用者在要求相同的資料時，網路節點經過查找之後會發現 Content Store 裡已經快取過一份了，因此，此要求可以直接從網路節點取得資料內容，節省 Interest 到資料提供者的時間。

IP 網路中的 WCCP 可以達到與 NDN 類似的快取功能，但兩者之間還是有些許的不同。WCCP 需要使用到其他的機器，去進行快取的動作。相較之下，NDN 是直接在網路節點上直接具備了一定的儲存空間，且並不需要設定。在這方面上，NDN 還是有著一定的優勢存在。比較圖二十一與圖二十二中的封包數，可看出 NDN 只需要一半的封包數即可完成相同的動作。

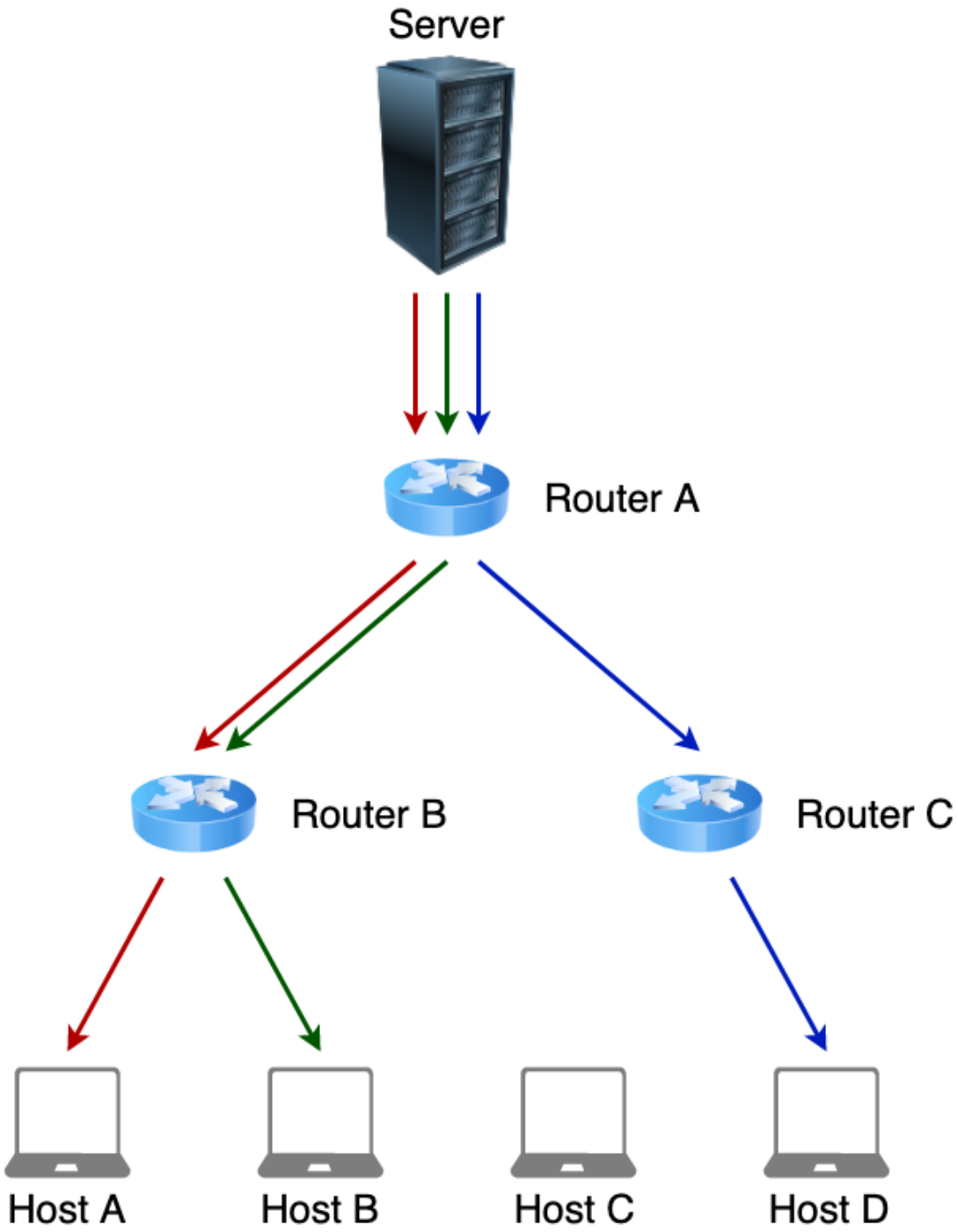
doi:10.6837/ncnu201900270

第四節 Multicast

第二節介紹了 NDN 的節點運作以及 Cache 與 Multicast 的功能；第三節介紹了 IP 網路的 Web Cache 協定及其運作流程；第四節將會介紹 IP 網路是如何運作 Multicast。

Multicast，多點廣播或者群播，是指將資料傳遞給一組目的位址 (在 Multicast 內稱為群組)。這樣的做法比起使用單點傳送（Unicast）將資料送到每一個目的位址，當然較有效率。因為來源只需要向群組 (Group) 發送一次資料封包的傳遞，在遇到不同的網路裝置 (Router 或者 Switch) 的時候，資料封包便會自動複製成多份，接著送到所有目的位址。下面會說明沒有使用 Multicast 與使用了 Multicast 的差別。

doi:10.6837/ncnu201900270

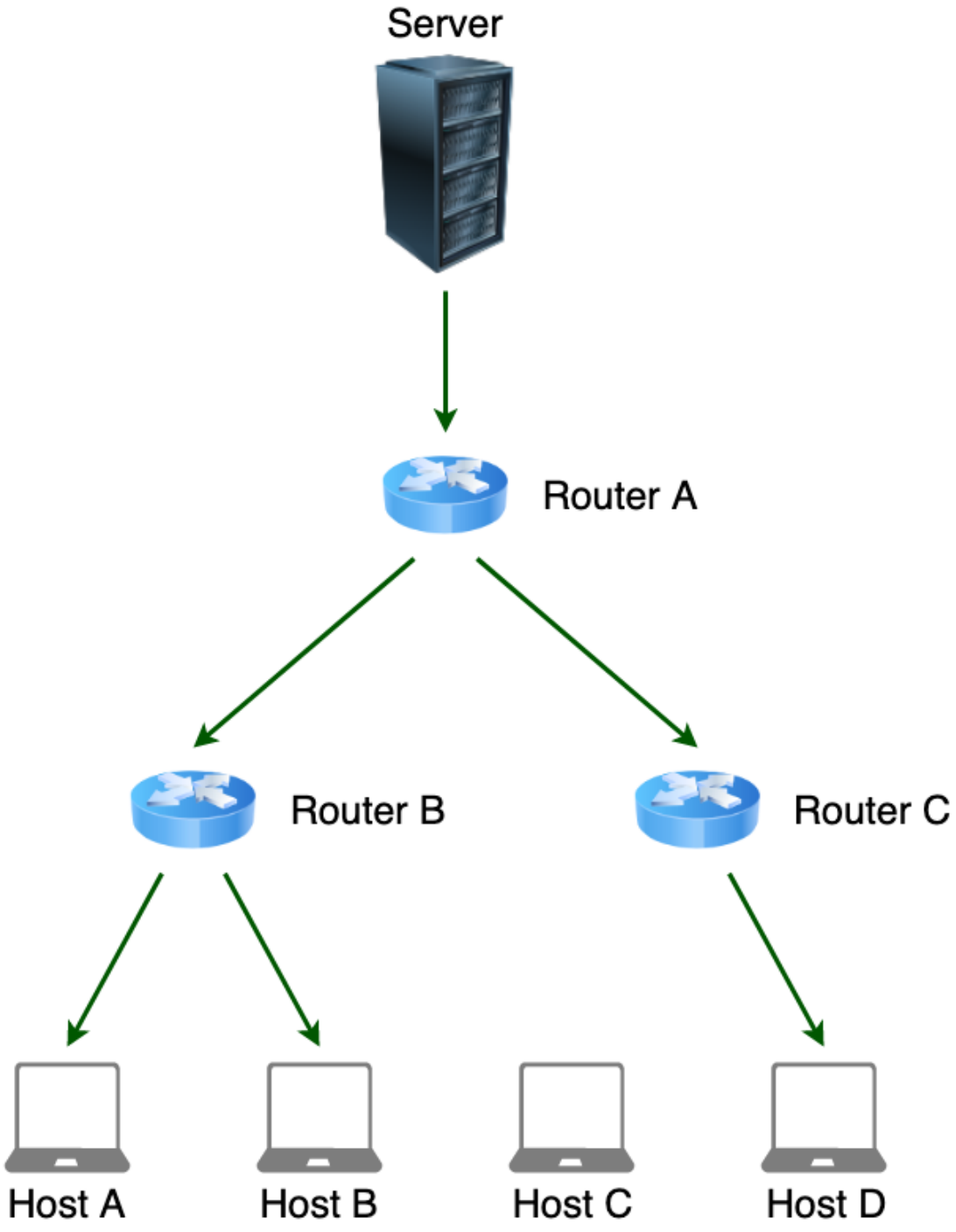


圖二十三、發送給多名使用者

傳統上在沒有使用 Multicast 的情況，當 Server 要將封包傳遞給 Host A、Host B、Host C 的時候，作法是準備多份內容相同，但目的位址不同的封包。送出時中間的路由會直接根據路由資料將封包送往目的位址，以此將封包傳遞給多

doi:10.6837/ncnu201900270

名使用者，如圖二十三所示。從圖上可以看出，這種做法的效率不是很好，因為相同的資料卻傳遞了三次。



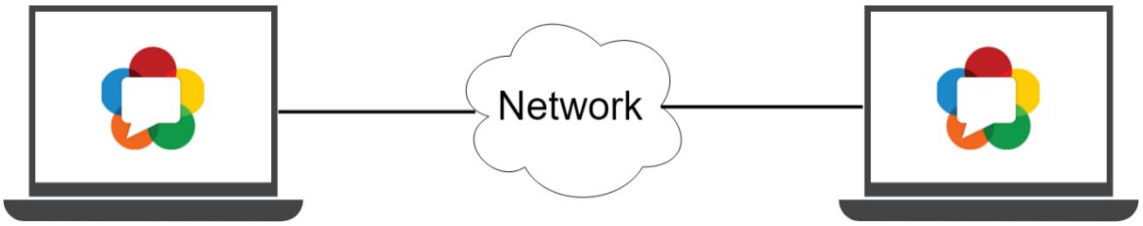
圖二十四、Multicast

相同情況下，若是使用 Multicast 的話， Server 只需發送一次封包，在遇到不同路徑的時候，路由器便會根據路由資料 (詳細 Multicast 的路由交換協定，請

doi:10.6837/ncnu201900270

參考 DVMRP [11]、MOSPF [12]、PIM [13])，將封包複製成多份並且將封包傳遞到目的位址 (Host A、Host B、Host C)，如圖二十四所示。從圖上即可發現，這種做法是比較有效率的，只需發送一次封包，伺服器的負擔自然減輕；中間的路由會自動將封包在需要時複製，減少了路徑上不必要的封包。

第五節 WebRTC



圖二十五、WebRTC

如圖二十五所示，Web Real-Time Communication [14] 最主要的概念就是，讓瀏覽器在不需要安裝程式或者外掛的情況下，達到即時的語音通訊或是視訊通訊。這部分會在本研究的網頁即時通訊應用展現。

第六節 技術挑戰

本研究參考了 NDN Browser，其如何以 NDN 作為底層傳輸去進行網頁的要求與網頁的呈現。在 Web Consumer 與 Web Producer 實作的部分，下面介紹到的部分會是本研究的技術挑戰。

1. NDN 協定的 URL 輸入：

   一般的瀏覽器，使用者可以透過輸入 URL 的方式來取得網頁，而 URL 的格式為：

   [協定] :// [位址] : [通訊埠] (ex: https://www.gazette.ncnu.edu.tw:443)

   協定有著其特定的功能，例如：HTTP 為瀏覽器與網頁伺服器進行溝通的協定；FTP 為瀏覽器與檔案伺服器進行文件傳輸的協定。在 NDN Browser 中，可以使用如 ccnx:// 的 URL 協定。本研究希望能在 Web Consumer 上進一步提供 ndn:// 的 URL 協定。

doi:10.6837/ncnu201900270

2. HTML 的解析：

取得 HTML 之後，瀏覽器會開始解析內容，其中，如 <script>、<img>、<audio>、<video> 等標籤 (tag) 可以使用 URL 作為來源，而一般的瀏覽器也僅支援 HTTP、HTTPS 等常見的協定。所以本研究希望加強 Web Consumer 在標籤的解析上，使其支援 NDN 協定。

另外，在設計網頁即時通訊應用上，可能會遇到下列的問題:

1. 命名：

由於 NDN 使用命名的方式是取得資料，因此，要去考慮用什麼樣形式來完成功能的設計。

2. 語音通訊:

IP 與 NDN 的封包運作機制差異相當大，在語音通訊方面，會不會因為運作方式不同而導致語音通訊的失敗。

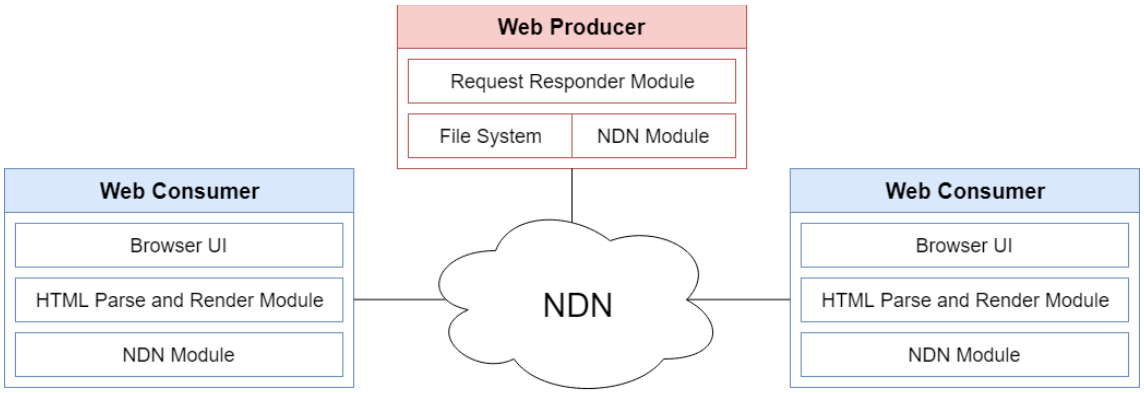
doi:10.6837/ncnu201900270

第三章 NDN 網頁即時通訊

第一節 系統架構

在傳統 IP 網路中，通常是由一台具備固定 IP 位址的伺服器負責提供服務，像是網頁伺服器、檔案伺服器以及 SIP 伺服器等。網頁即時通訊(WebRTC)服務結合了網頁伺服器與 SIP 伺服器，使用者經由瀏覽器向網頁伺服器取得即時通訊應用的程式碼之後，接著此應用便會開始與 SIP 伺服器溝通。此種做法可以讓我們利用瀏覽器取得應用頁面，並且與其他使用者進行語音通訊。這種 Client/Server 架構雖然便利，但也產生了一些問題，例如: 當使用者增加，伺服器的負擔就會提高；若是超過上限，甚至可能會導致整個服務的崩潰。此外，提供服務會使得大量的封包流進流出伺服器，這也有可能造成伺服器附近的路由嚴重阻塞。在本章中接下來除了會介紹 Web Producer 與 Web Consumer 之外，同時也會提到 NDN 是如何達到減輕服務提供者的負擔以及達成點對點的溝通。

第二節 Web Producer 模組介紹



圖二十六、系統架構圖

圖二十六為系統架構圖。在 NDN 中，Web Producer 負責提供網頁內容；Web Consumer 負責要求網頁並且與使用者互動。

doi:10.6837/ncnu201900270

第一小節 Web Producer

Web Producer 負責回應網頁的要求，在 NDN 中註冊的 Prefix 為 /ncnu/chat，並提供 index.html、js/main.js、js/ndn.js、js/RecordRTC.js 及 js/defaultRsa.js 等，NDN 網頁即時通訊應用所需的檔案。

Web Producer 具備了三個模組，模組功能介紹如下。

Request Responder Module:

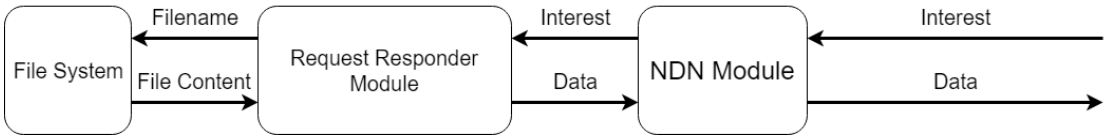
判斷從 NDN Module 收到的 Interest，此模組會分析 Interest 的 Name 所要求的內容為何。接著，再向 File System 要求此檔案並且根據序列號碼決定 Data 的 Content。

File System:

當 Request Responder Module 分析完 Interest 的 Name 所要求的內容後，會由 File System 從指定的資料夾目錄中提取檔案，並交由 Request Responder Module 處理。

NDN Module:

NDN 的網路模組，負責註冊 Prefix 及收送 Interest 與 Data 封包。



圖二十七、Web Producer 運作流程

NDN Module 會在 NDN 網路節點中註冊 Prefix (/ncnu/chat)，接著 NDN 網路節點會將匹配到該 Prefix 的 Interest 送至 Web Producer。

Web Producer 運作流程如圖二十七。在接收到 Interest 之後，會將 Interest 交給 Request Responder Module，接著，開始分析 Interest 的 Name 所要求的內容為何。例如: /ncnu/chat/index.html/0；其中，/ncnu/chat 為本應用在 NDN 註冊的 Prefix，index.html 為此 Interest 所要求的檔案名稱，0 則為 index.html 被切割成多

doi:10.6837/ncnu201900270

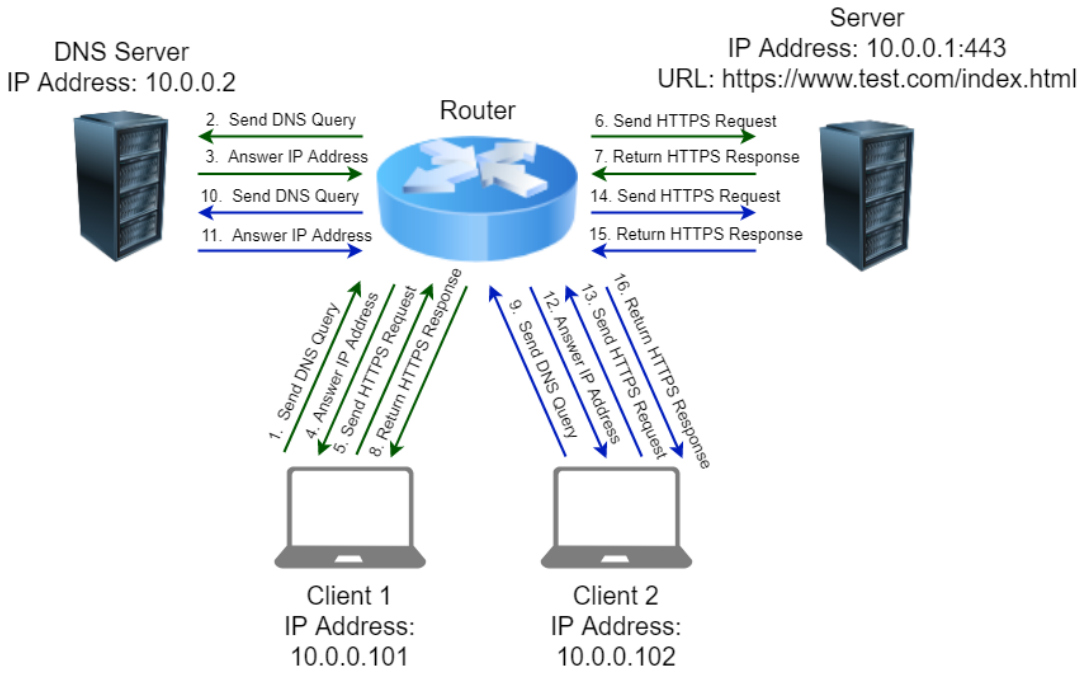
個小片段的序列號碼。分析完成之後，Request Responder Module 便會向 File System 要求檔案。

此時，File System 會收到 Request Responder 所要求的檔案名稱，並從當前目錄去提取檔案，再將檔案內容交給 Request Responder Module 處理。取得檔案內容後，Request Responder Module 便會建立一個與 Interest 的 Name 相同的 Data，並根據 Interest 的要求，將檔案內容放進 Content 之中。

最後，NDN Module 會收到 Request Responder Module 所建立好的 Data，並將其回傳給 Consumer。

第二小節 IP 與 NDN 網頁存取服務差異

在 IP 網路中，每台裝置都必須具備 IP 位址，這樣才能與其他的裝置溝通。此外，欲取得資訊，除了 IP 位址之外，還必須指定完整的 URL（Uniform Resource Locator），進一步指定在伺服器上哪個目錄下的哪個檔案。



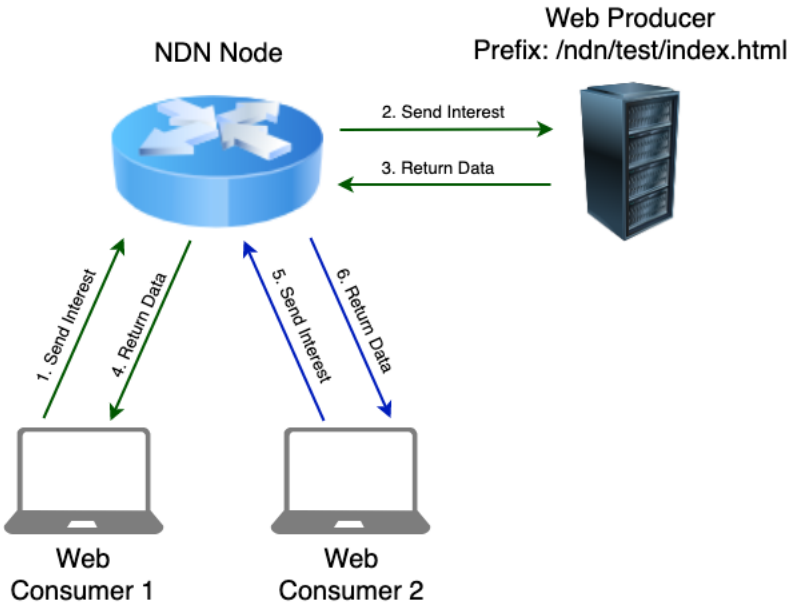
圖二十八、網頁要求流程圖 (IP 網路)

使用者 (Client 1) 瀏覽網頁的方式，如圖二十八所示。在輸入 URL 後，瀏覽器會先到 DNS 伺服器 (Step 1~2) 查詢，取得 URL 中伺服器主機 所對應的 IP 位址 (Step 3~4)，接著再向網頁伺服器進行要求 (Step 5~6)。網頁伺服器在收到要求

doi:10.6837/ncnu201900270

後，便會回傳其所要求的網頁內容 (Step 7~8)。瀏覽器接收到內容後，再由畫面上顯示出來。

當有另外一名使用者 (Client 2) 也想要求相同網頁的時候 (Step 9~16)，使用者仍需先向 DNS 伺服器發出查詢，接著才依查到的 IP 位址向網頁伺服器進行要求。



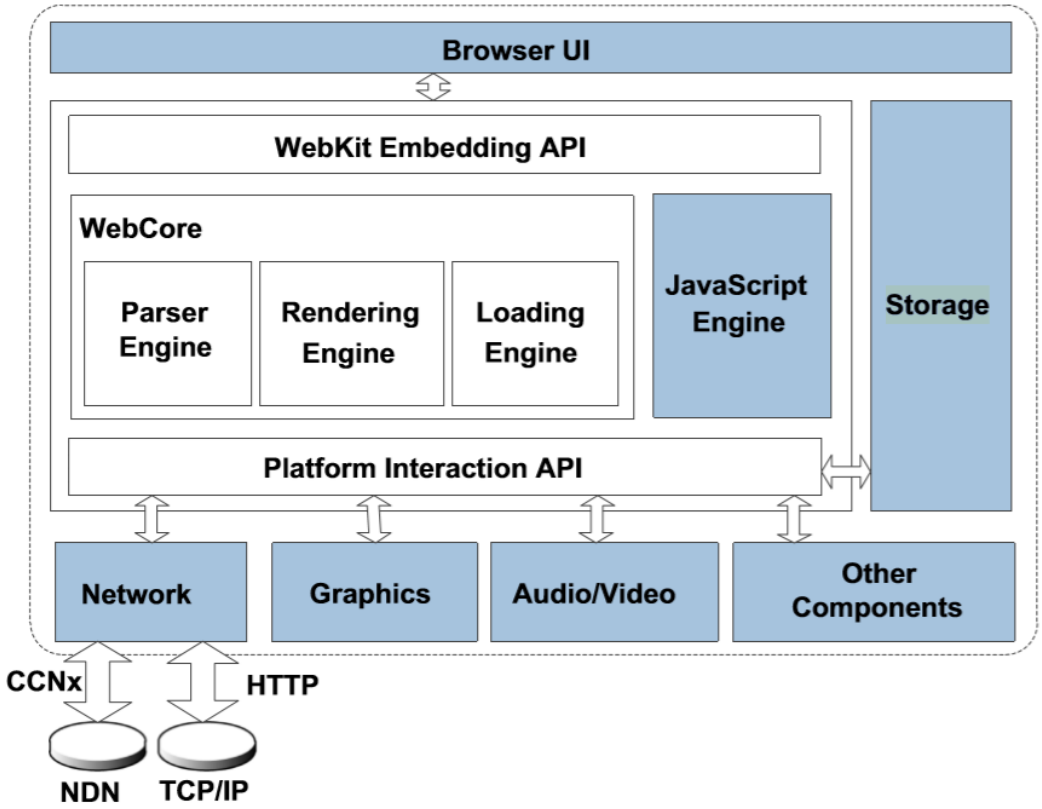
圖二十九、網頁要求流程圖 (NDN 網路)

相對地，NDN 以命名資料取代 IP 位址，因此，在要求網頁的時候，如圖二十九所示。使用者 (Web Consumer 1) 只要輸入 NDN 協定的 URL，Web Consumer 便會送出 Interest 去要求網頁，其中，Interest 的 Name 會匹配到 Web Producer (網頁提供者) 註冊的 Prefix。NDN Node 會根據匹配結果，直接將 Interest 送至 Web Producer (Step 1~2)，這過程節省了訪查 DNS 伺服器的時間與流量，接著 Web Producer 的 Data 會沿著 Interest 的路徑回傳 (Step 3~4)，最後由瀏覽器顯示網頁內容。

與 IP 網路要求網頁的運作不同，當有另外一名使用者 (Web Consumer 2) 也想要求相同網頁的時候，使用者可以在直接從路徑上已經快取過資料的節點取得網頁內容 (Step 5~6)。不僅不需要向 DNS 伺服器查詢，同時也展現了 NDN 其中一個重要的特色，快取機制，達到減少伺服器負擔與流量的效果。

doi:10.6837/ncnu201900270

第三小節 NDN Browser 模組介紹



圖三十、NDN Browser 架構

如本論文的研究目的所述，CCNx 為 NDN 稍早版本所使用的網路環境。NDN 在 2013 年的時候，從 CCNx 其中一個分支上，開發出符合 NDN 要求的網路環境 NFD。而目前網路上所流傳的 NDN Browser， 所使用的是已被 NFD 所取代的 CCNx。因此，本研究在 Web Consumer 底層傳輸的部分，改為使用 NDN 的 NFD，以達成 NDN 的封包傳輸。下面會先介紹到 NDN Browser 的哪些部分使用了 CCNx，接著會說明 Web Consumer 如何將這些部分改為使用 NFD。圖三十為 NDN Browser 的架構。圖中藍色的部分代表會使用到 CCNx 的區塊。

Browser UI:

提供使用者輸入 CCNx 協定的 URL，並且顯示網頁的輸出畫面。

Network:

為網路的區塊，負責收送 CCNx 以及 IP 的封包 (NDN Browser 亦可瀏覽 HTTP 的網頁)。

doi:10.6837/ncnu201900270

Graphics:

HTML 中的 <img>，來源可為 CCNx 協定的 URL。

Audio/Video:

HTML 中的 <audio> 與 <video>，來源可為 CCNx 協定的 URL。

Storage:

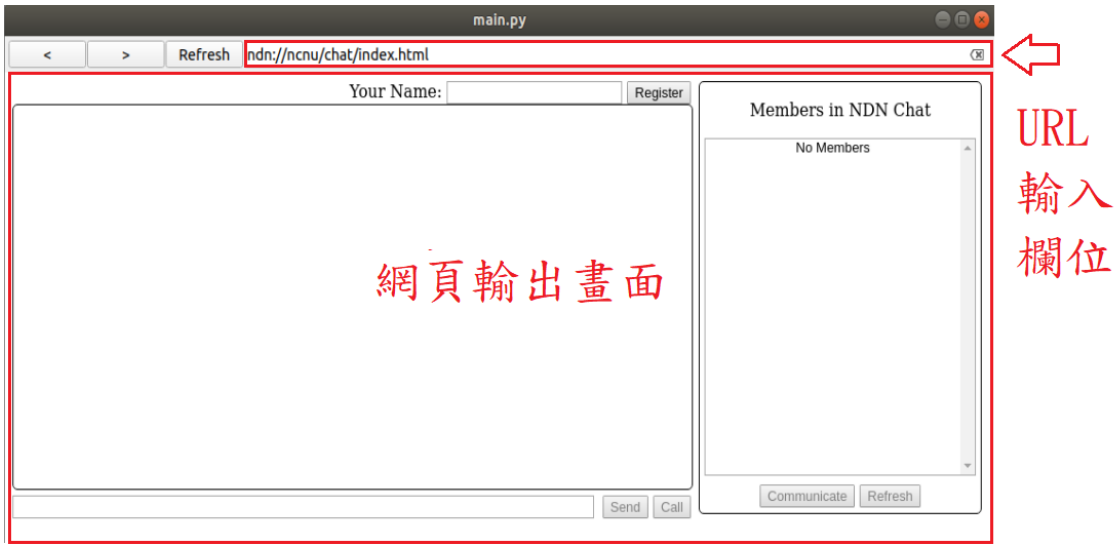
儲存資料的區域。

JavaScript Engine:

提供 CCNx 物件的使用。

Network、Graphics、Audio/Video 會透過 CCNx 取得網頁的資料內容，並且儲存在 Storage 內，再透過 WebCore 的 Parser Engine、Rendering Engine 以及 Loading Engine 將資料讀入並且建構輸出畫面，最後由 Browser UI 顯示網頁。

第四小節 Web Consumer 模組介紹



圖三十一、Web Consumer

本研究中的 Web Consumer 是以 PyQt5 [15] 實作的應用程式，其介面及 NDN 網頁即時通訊應用，如圖三十一所示。Web Consumer 負責網頁的要求與顯示，具備三個模組：

1.Browser UI:

doi:10.6837/ncnu201900270

參考 NDN Browser 其中 Browser UI 的部分，提供 NDN 協定的 URL 輸入以及網頁的顯示。

2.HTML Parse and Render Module:

參考 NDN Browser 其中 Graphics、Audio/Video、Storage 以及 JavaScript Engine，將這些部分整合而成此 Module。負責解析 HTML 以及建構輸出畫面，並且支援標籤來源為 NDN 協定的 URL。

3.NDN Module:

參考 NDN Browser 其中 Network 的部分，為 Web Consumer 的 NDN 網路模組，負責收送 Interest 與 Data 封包。

第五小節 Web Consumer 模組運作說明

在 Browser UI 中輸入 NDN 協定的 URL，接著透過 NDN Module 向 Web Producer 要求網頁內容，例如：要求本研究的 NDN 網頁即時通訊應用是輸入 ndn://ncnu/chat/index.html。NDN Module 會送出 Interest /ncnu/chat/index.html/0、/ncnu/chat/index.html/1 等，index.html 為此 Interest 所要求的檔案名稱，0 則為 index.html 被切割成多個小片段的序列號碼。

當收到 Web Producer 回傳的 Data 後，由於會收到多個 index.html 的小片段，Web Consumer 會先將 index.html 進行重組，接著讀入。此時， HTML Parse and Render Module 會開始進行 HTML 的解析。

HTML Parse and Render Module 支援標籤來源為 NDN 協定的 URL，例如：本研究的 NDN 網頁即時通訊應用，其 HTML 會包含

<script src="ndn://ncnu/chat/js/main.js"></script>

表示 main.js 需要透過 NDN 取得。

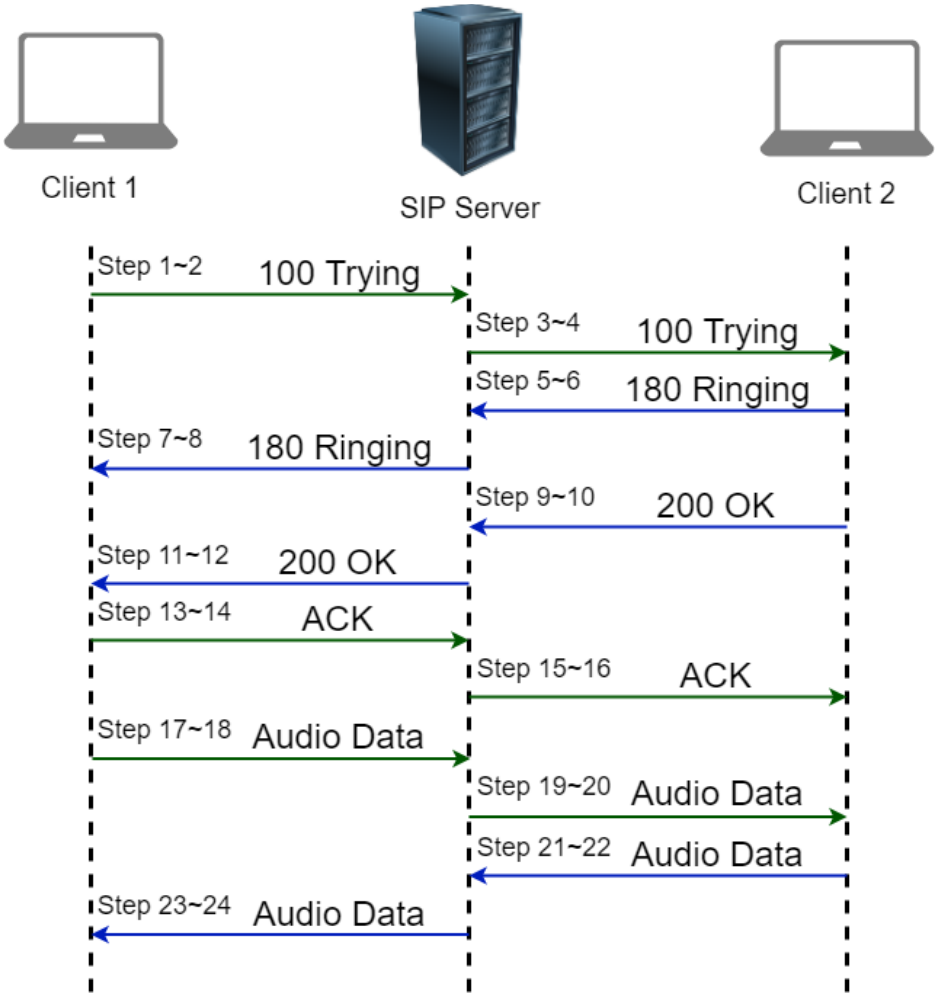
此時，HTML Parse and Render Module 會透過 NDN Module 向 Web Producer 要求資料，如：ndn://ncnu/chat/js/main.js 會送出 Interest /ncnu/chat/js/main.js/0、/ncnu/chat/js/main.js/1 等。Web Producer 回傳的 Data，亦會是多個的小片段，因此，HTML Parse and Render Module 會先將檔案重組，然後讀入。

doi:10.6837/ncnu201900270

在收齊所有資料之後，HTML Parse and Render Module 才會完成網頁的建構，最後由 Browser UI 將網頁顯示出來。

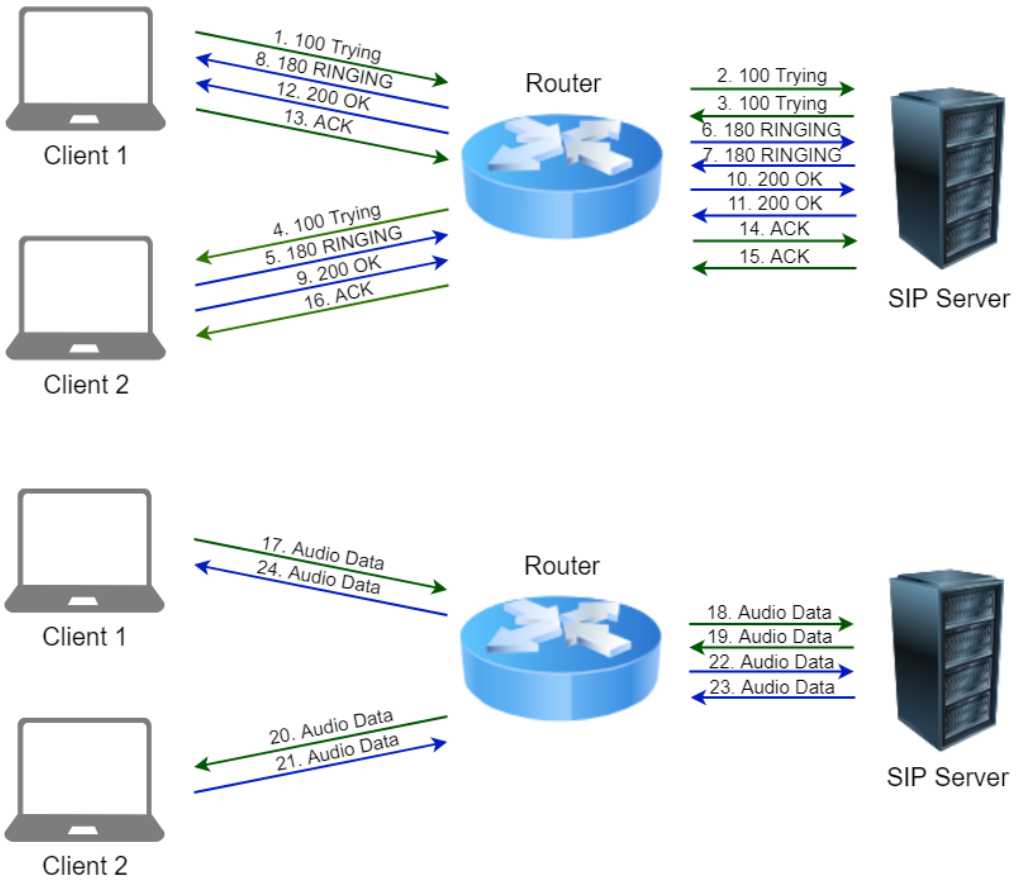
第三節 應用架構

第一小節 IP 與 NDN 差異



圖三十二、即時通訊信令流程 (IP 網路)

doi:10.6837/ncnu201900270

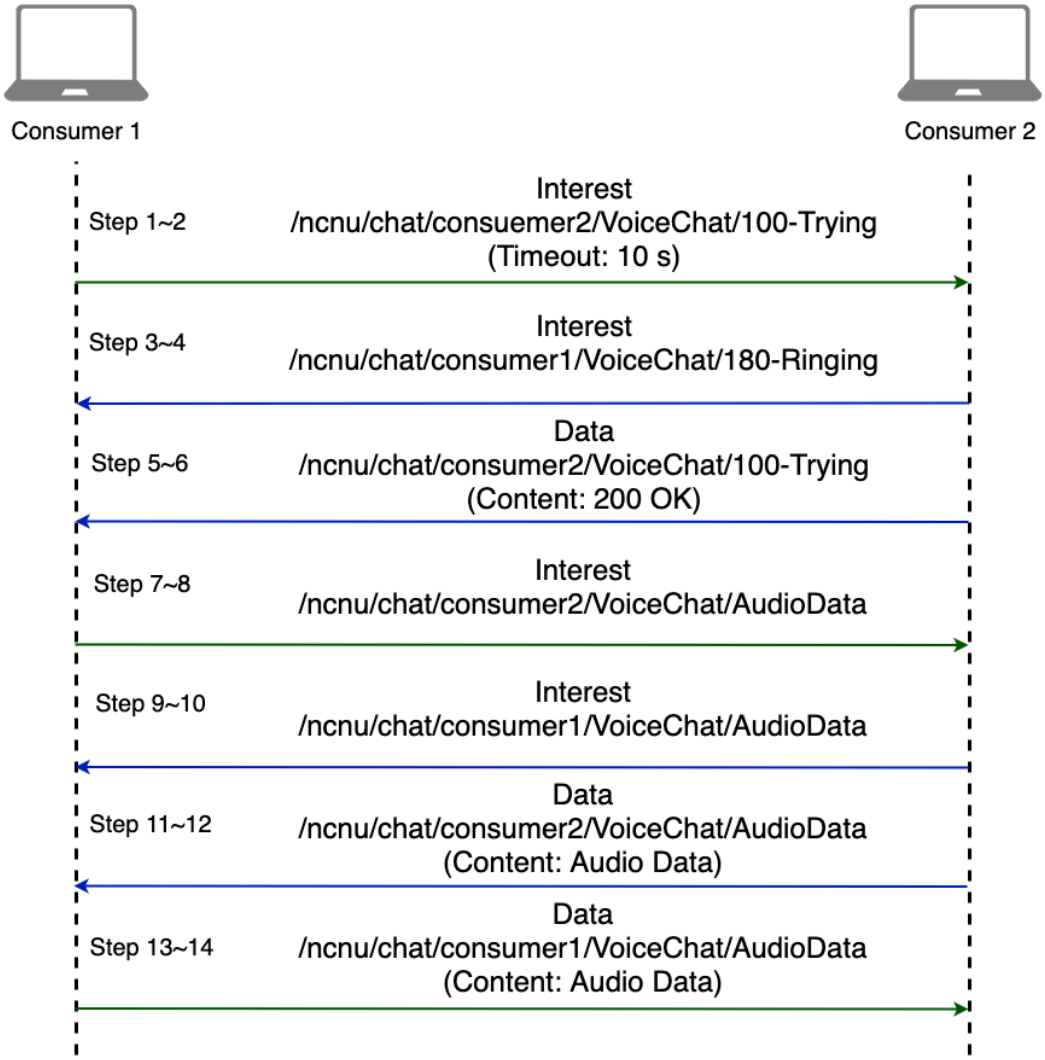


圖三十三、即時通訊資料走向圖 (IP 網路)

圖三十二與圖三十三所示為 IP 網路中，即時通訊的信令流程以及實際的封包在路由器上的走向圖。從圖中可以發現，在取得應用之後，仍需要由伺服器作為兩者之間溝通的橋梁。Client 1 會向 SIP Server 發送 100 Trying，要求建立語音通訊的會話 (Session)，等待過程會從 SIP Server 收到 Client 2 的 180 Ringing，告知已振鈴但尚未接聽。當 Client 1 從 SIP Server 收到 Client 2 回應的 200 OK 後，即表示會話已成功建立。接著，兩者會經由 SIP Server 向對方發送語音訊息的資料，完成語音通訊。

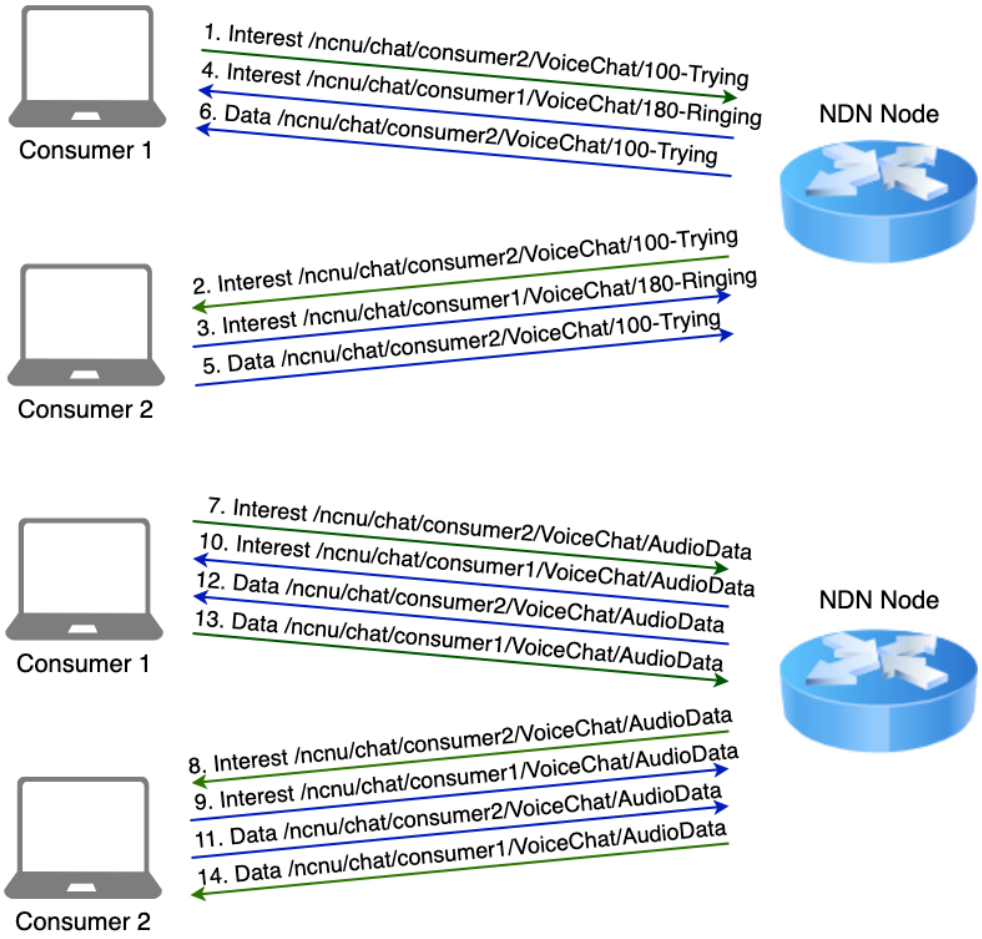
傳統上，兩個使用者的設備可能因 IP 浮動、NAT 穿越等問題，不易直接進行點對點的溝通。因此，以一台具備固定 IP 位址的伺服器將兩者連接起來是被廣為採用的解決方案。但這也造成了伺服器的負擔以及其附近路由會有大量封包等問題。

doi:10.6837/ncnu201900270



圖三十四、即時通訊信令流程 (NDN)

doi:10.6837/ncnu201900270



圖三十五、即時通訊流程圖 (NDN)

NDN 以命名資料取代 IP 位址，加上網路節點可以直接經由 Name 找到資料提供者 (Producer)，因此不會有 IP 位址浮動以及穿越 NAT 等問題，可以輕鬆達到點對點的溝通。

首先，Consumer 1 與 Consumer 2 分別會在 NDN 網路節點上註冊 /ncnu/chat/consumer1/VoiceChat 以及 /ncnu/chat/consumer2/VoiceChat 的 Prefix，當 NDN 網路節點收到 Interest 後，就會根據匹配 Prefix 的結果，將 Interest 送至 Consumer 1 或 Consumer 2。

如圖三十四以及圖三十五所示，信令的使用參考 [16]。Consumer 1 會送出 100 Trying 的信令，也就是 Interest /ncnu/chat/consumer2/VoiceChat/100-Trying，向 Consumer 2 發起語音通訊的會話 (Session) 要求。若 Comsumer 1 在 Interest 送出 10 秒後 (超過 Interest 的 Lifetime) 沒得到 Data，會判斷該 Interest 已經 Timeout，代表 Consumer 2 沒有回應。

doi:10.6837/ncnu201900270

Consumer 2 收到 Interest /ncnu/chat/consumer2/VoiceChat/100-Trying 後，會先送出 180 Ringing 信令 (Interest /ncnu/chat/consumer1/VoiceChat/180-Ringing)，告知已振鈴但尚未接聽。

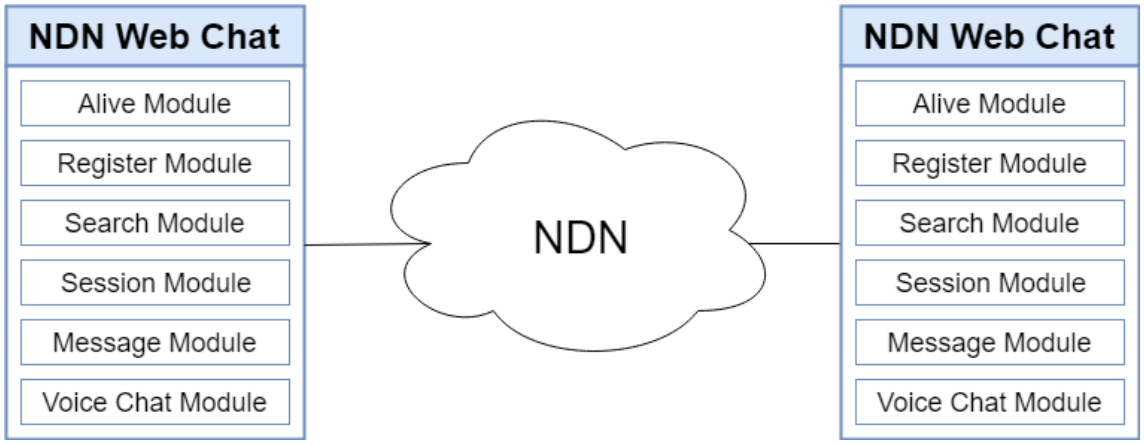
接著會有 10 秒的時間，可以回應此 Interest。當 Consumer 2 接受語音通訊要求，會回應 Data，Name 為 /ncnu/chat/consumer2/VoiceChat/100-Trying，Content 則是 200 OK。

Consumer 1 在收到 Content 為 200 OK 的 Data 後，即可得知 Consumer 2 接受了語音通訊的要求。

兩者在確認會話建立後，便會開始向對方要求語音訊息的資料，完成語音通訊，例如：Consumer 1 發送 Interest /ncnu/chat/consumer2/VoiceChat/AudioData 向 Consumer 2 要求語音訊息的資料。

在這過程可以看到，無需伺服器協助的情況下，兩者可直接完成會話的建立與聲音資料的傳輸。

第二小節 NDN 網頁即時通訊應用



圖三十六、NDN 網頁即時通訊應用架構

在簡單介紹完 IP 網路與 NDN ，兩者之間的使用差異之後，接著會說明 NDN 網頁即時通訊應用的模組功能。圖三十六中顯示了 NDN 網頁即時通訊應用所需要的模組，參考現有的 [17][18][19]，並且使用 [20] 所提供的應用程式介面完成。

doi:10.6837/ncnu201900270

NDN 網頁即時通訊應用總共有六個模組，描述如下：

1. Alive Module：

   判斷是否會重複註冊相同 ID

2. Register Module：

   註冊 ID

3. Search Module：

   搜尋已註冊的使用者

4. Session Module：

   建立會話

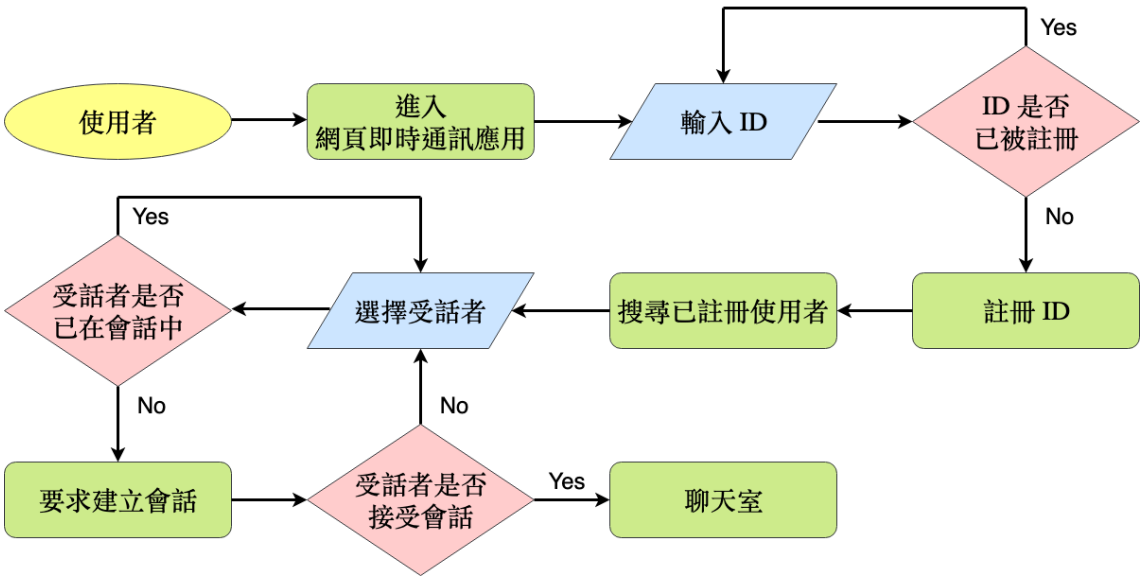
5. Message Module：

   文字訊息交換

6. Voice Chat Module ：

   語音通訊

第四節 操作流程



圖三十七、NDN 網頁即時通訊應用流程圖

圖三十七為 NDN 網頁即時通訊應用的流程圖，接著搭配流程圖說明整個操作流程。首先，使用者 (流程圖中的 Consumer) 在進入網頁即時通訊應用之後

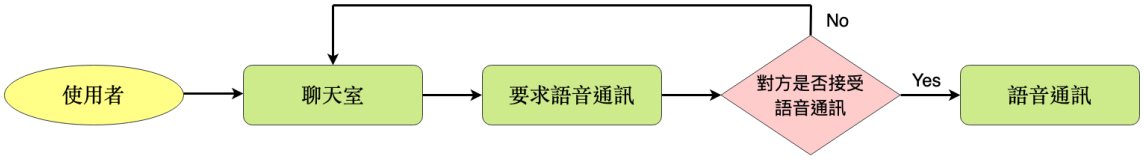
doi:10.6837/ncnu201900270

會需要輸入 ID，在註冊 ID 之前，本應用的 Alive Module 會先檢查此 ID 是否已被註冊。

確認 ID 可被使用之後，本應用的 Register Module 才會以此 ID 進行註冊。完成註冊後，使用者即可透過 Search Module，搜尋其他的線上使用者。

接著，使用者可以在搜尋到的名單中，選擇一名受話者，要求會話 (Session) 的建立，本應用的 Session Module 會先檢查該受話者是否已在會話中。

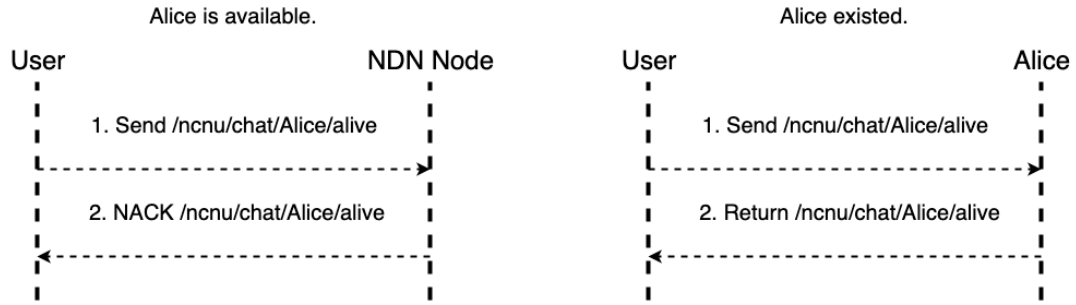
確認該受話者不在會話中後，Session Module 才會向該受話者發起會話建立的要求。受話者可以選擇是否接受會話的建立，若是接受，雙方便可進入聊天室。



圖三十八、要求語音通訊流程圖

圖三十八為進入聊天室後，進一步要求語音通訊的流程圖。雙方在完成會話的建立後，會進入到聊天室，能透過 Message Module 進行文字訊息的交換。除此之外，雙方亦能使用 VoiceChat Module 向對方發起語音通訊的邀請，若是對方允許的話，便會進入語音通訊的部分。

第一小節 Alive Module



圖三十九、Alive Module 信令流程

圖三十八為 Alive Module 的訊號流圖。假設，使用者想註冊 Alice 這個 ID。此時，Alive Module 會送出 Interest /ncnu/chat/Alice/alive 的 Interest。若是已有人註冊了 Alice 作為 ID，只要使用者收到 Data /ncnu/chat/Alice/alive，便可得知

doi:10.6837/ncnu201900270

Alice 已被註冊的訊息。若是收到 NACK，則表示此 ID 沒有人使用。因為並沒有人註冊 Prefix /ncnu/chat/Alice/alive，所以 NDN Node 無法將此 Interest 送出，此時，NDN Node 就會回傳 NACK，告知該資料不存在。

第二小節 Register Module

當確認完想註冊的 ID 為可以使用之後，本應用便會執行 Register Module，在 NDN 網路節點中，為此使用者註冊以下幾個 Prefix。

1. /ncnu/chat/user_id/alive

   為 Alive Module 所使用的 Prefix，用於回應此 user_id 已被此使用者註冊。

2. /ndn/broadcast/ncnu/chat/members

   為 Search Module 所使用的 Prefix，用於回應此使用者的 ID。

   此外，/ndn/broadcast 為廣播機制的 Prefix，當 NDN 網路節點收到 Interest，Name 為 /ndn/broadcast/ncnu/chat/members，NDN 網路節點會將 Interest 送至所有註冊此 Prefix /ndn/broadcast/ncnu/chat/members 的使用者。若是沒有廣播機制，則會發生註冊相同 Prefix 的使用者，只有第一個註冊的使用者會收到 Interest。

3. /ncnu/chat/user_id/session

   為 Session Module 所使用的 Prefix，用於回應會話的要求。

4. /ncnu/chat/user_id/message

   為 Message Module 所使用的 Prefix，用於接收文字訊息。

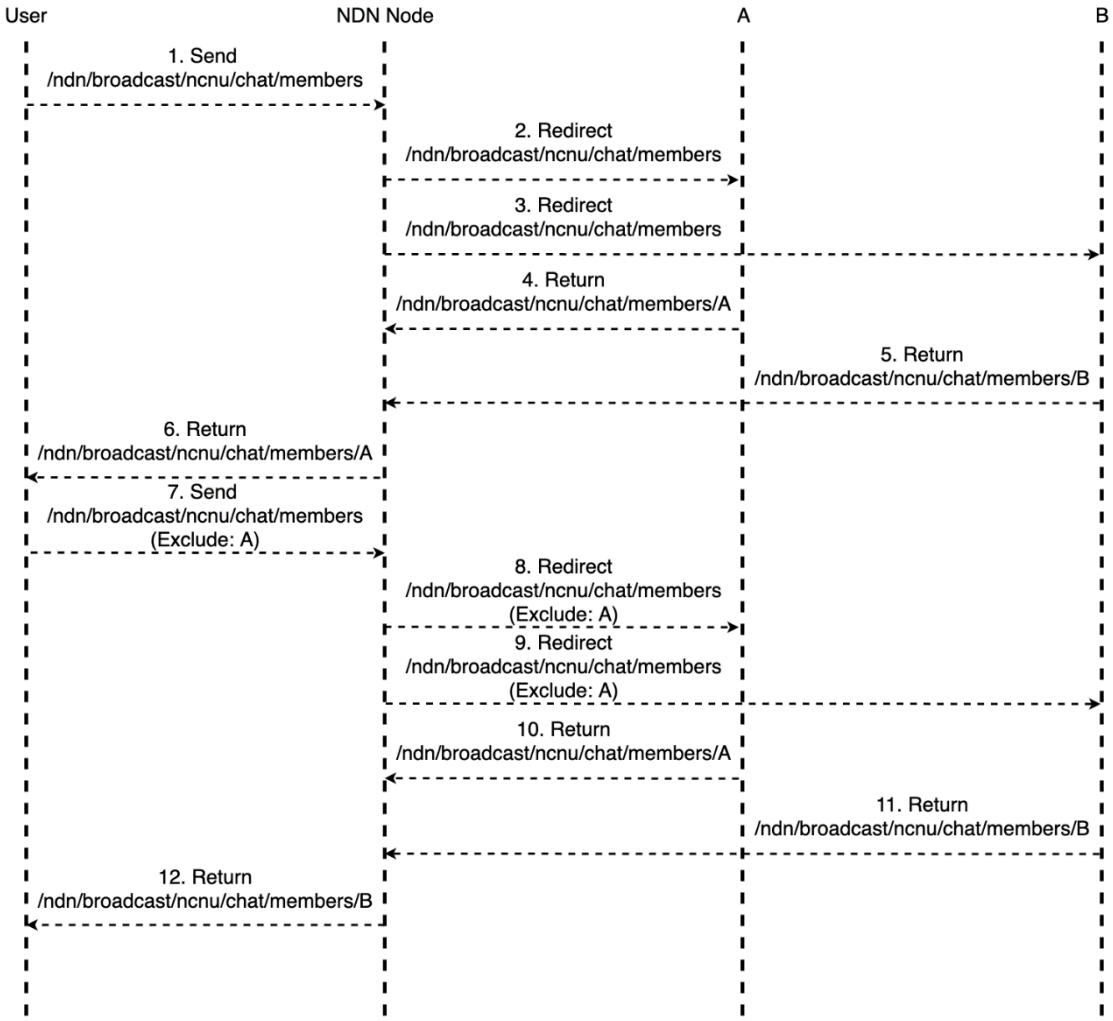
5. /ncnu/chat/user_id/VoiceChat

   為 VoiceChat Module 所使用的 Prefix，用於回應語音通訊的要求，以及聲音資料的內容。

第三小節 Search Module

完成註冊之後，使用者即可利用 Search Module 的功能，搜尋本應用的所有線上使用者。並且由 Session Module 與他們進行會話的建立。

doi:10.6837/ncnu201900270



圖四十、Search Module 信令流程

Search Module 為搜尋本應用內所有線上使用者的功能。由於 NDN 為一送一收的機制，儘管使用 /ndn/broadcast 也無法完整地得知應用內的所有線上使用者。因爲每個人都會回覆，但 NDN Node 只回傳最先收到的 Data，這就導致了重複回應的問題。所以，Search Module 在 Interest 上加入了 Exclude (為 Interest 的一項參數，負責記錄已回應過 Data 的使用者)，區別已經得知的使用者。

在圖四十的流程中，A 跟 B 都註冊 Prefix /ndn/broadcast/ncnu/chat/member。當 User 想要搜尋本應用下的使用者時，會使用 Search Module。

首先，在送出第一次 Interest /ndn/broadcast/ncnu/chat/members 後，NDN Node 會根據 /ndn/broadcast，得知這是廣播的封包，並且匹配 A 跟 B 的 Prefix，因此，轉送 Interest 至 A 跟 B。接著，A 跟 B 會將自己的 ID 添加在 Data 的 Name

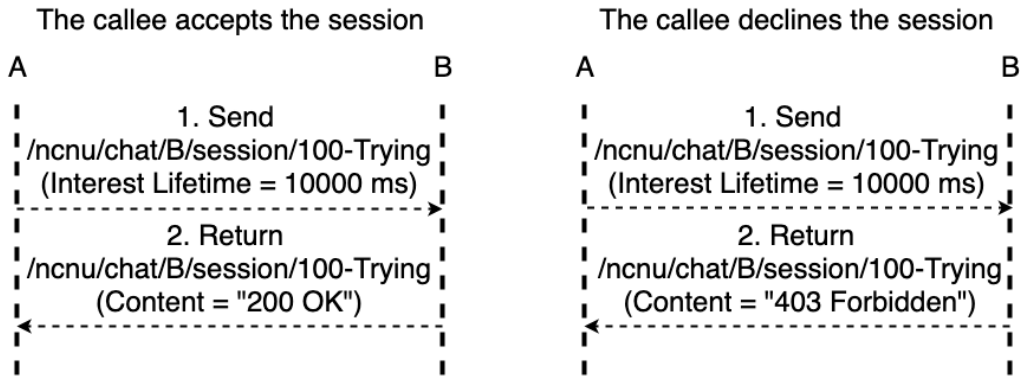
doi:10.6837/ncnu201900270

後，如：Data /ndn/broadcast/ncnu/chat/members/A，然後回傳。在第一次的搜尋中，A 先回應了，因此，User 只會收到 A 的 Data，B 的 Data 則是被 NDN Node 丟棄 。

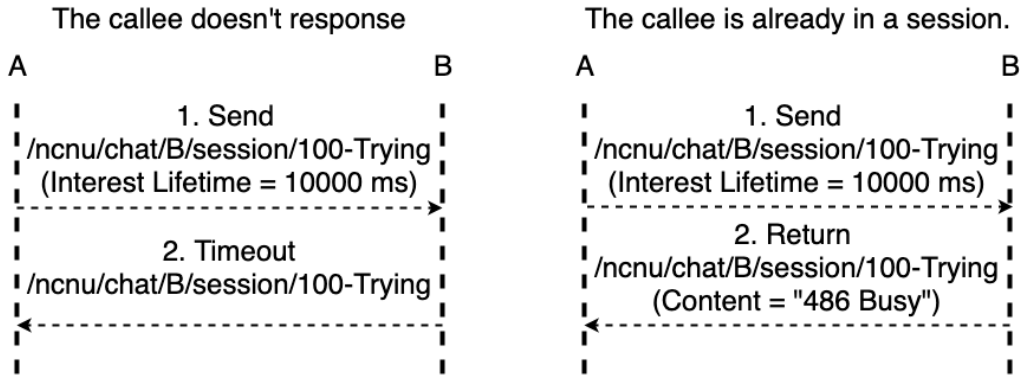
接著，送出第二次 Interest /ndn/broadcast/ncnu/chat/members 並且在 Exclude 加入已經搜尋到的 “A”，NDN Node 依然會將 Interest 轉送至 A 跟 B。而 A 跟 B 也仍然會回傳 Data，但當 Data 到達 NDN Node 的時候，NDN Node 會根據 Interest 上的 Exclude，判斷 A 是在被排除的清單中，因此，將 B 的 Data /ndn/broadcast/ncnu/chat/members/B 回傳給 User，A 的 Data 則是丟棄。

在這樣的運作流程下，只要持續更新 Interest 的 Exclude，就能得知本應用內的所有線上使用者。

第四小節 Session Module



圖四十一、Session Module 信令流程 (接受、拒絕)



圖四十二、Session Module 信令流程 (沒有回應、已在會話中)

圖四十一為 Session Module 的信令流程。使用者在搜尋完本應用的使用者名單後，可以選擇一名受話者 (callee)，與其建立會話。

doi:10.6837/ncnu201900270

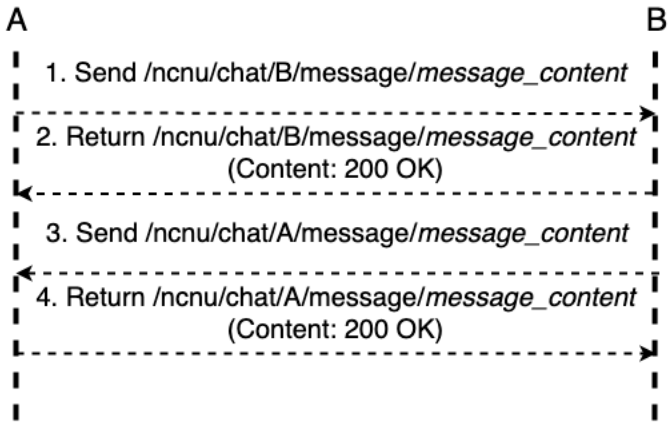
假設有 A 跟 B 兩個使用者，雙方都註冊了 Session Module 的 Prefix，(A 為 /ncnu/chat/A/session，B 為 /ncnu/chat/B/session)，而此次是由 A 向 B 發起會話的建立，A 為發話者 (caller)，B 為受話者 (callee)。

A 會透過 Session Module 向 B 送出 Lifetime 為 10000 ms (10秒) 的 Interest，Name 為 /ncnu/chat/B/session/100-Trying。B 在收到 Interest 後，有 10 秒的時間決定是否接受。

B 所回應的 Data，Name 為 /ncnu/chat/B/session/100-Trying，Content 則可填入 200 OK 或者 403 Forbidden。200 OK 代表 B (callee) 接受會話建立，403 Forbidden 則代表 B (callee) 拒絕會話建立。

若 A 在10秒內沒有收到 B 回傳的 Data，則會判斷該 Interest 已經 Timeout 了，代表 B 沒有選擇接受。此外，還有 B 已經在會話中的可能，此時 B 會直接回應 Data，其 Content 內容為 486 Busy。

第五小節 Message Module



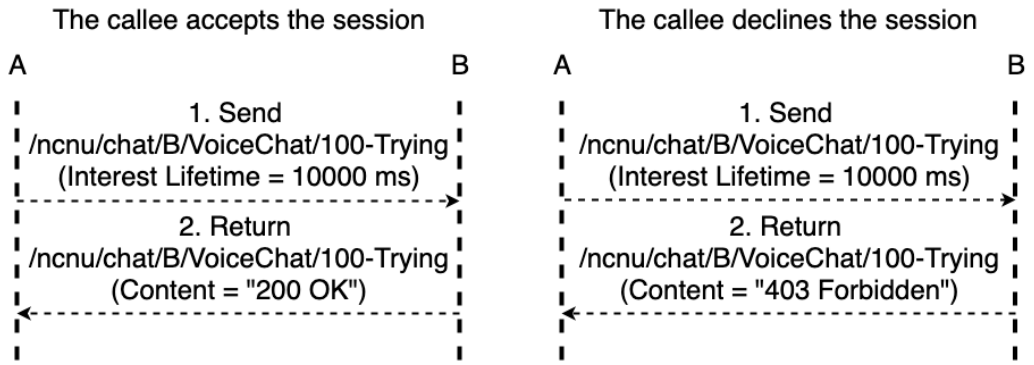
圖四十三、Message Module 信令流程

圖四十三為 Message Module 信令流程。當受話者接受會話 (Session) 的建立之後，發話者與受話者便會進入到聊天室。雙方都已註冊了 Message Module 的 Prefix (A 為 /ncnu/chat/A/message，B 為 /ncnu/chat/B/message)，假設 A 要向 B 送出訊息 (Hello)，A 會在對話框之中輸入訊息，接著 Message Module 會將訊息內容 (*message_content*) 添加在 Interest 的 Name 後方，如：/ncnu/chat/B/message/Hello，然後送出 Interest。

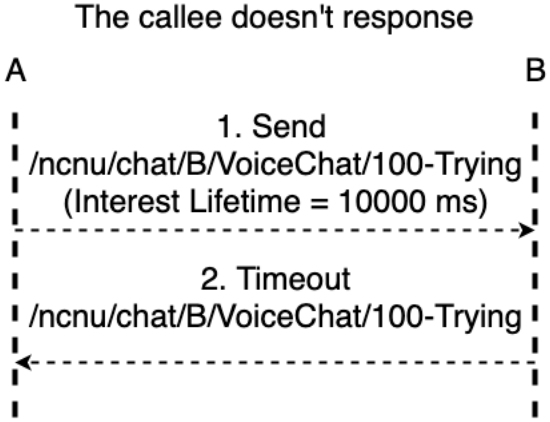
doi:10.6837/ncnu201900270

當 B 收到 Interest 之後，則會回應 Data，Name 為 /ncnu/chat/b/message/Hello，Content 為 200OK，表示已經收到訊息了。

第六小節 VoiceChat Module

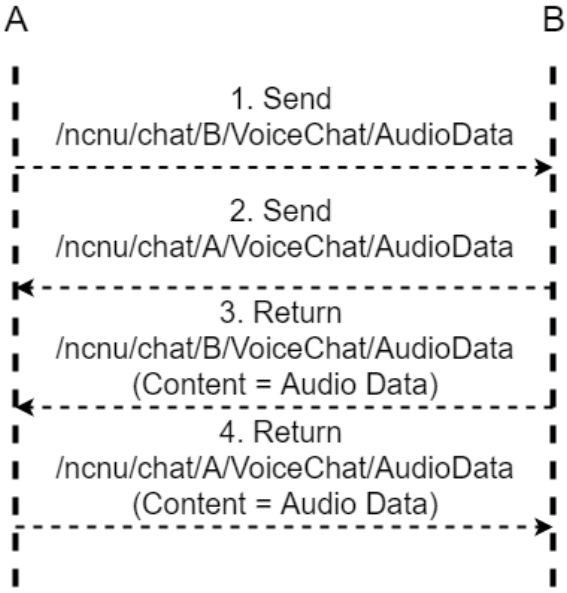


圖四十四、VoiceChat Module 信令流程 (接受、拒絕)



圖四十五、VoiceChat Module 信令流程 (沒有回應)

doi:10.6837/ncnu201900270



圖四十六、VoiceChat Module 信令流程 (聲音的傳送)

完成會話的建立之後，除了文字訊息交換的功能，使用者還可以使用 VoiceChat Module 的功能，邀請對方進行聲音的資料交換，以進行兩者之間的語音通訊。

與 Session Module 相同，有 A 跟 B 兩個使用者，且雙方都註冊了 VoiceChat Module 的 Prefix，(A 為 /ncnu/chat/A/VoiceChat，B 為 /ncnu/chat/B/VoiceChat)。此次是由 A 向 B 發起語音通訊的邀請，A 為發話者 (caller)，B 為受話者 (callee)。

A 會透過 VoiceChat Module 向 B 送出 Lifetime 為 10000 ms (10秒) 的 Interest，Name 為 /ncnu/chat/B/VoiceChat/100-Trying。B 在收到 Interest 後有 10 秒的時間決定是否接受。

B 所回應的 Data，Name 為 /ncnu/chat/B/VoiceChat/100-Trying，Content 則可填入 200 OK 或者 403 Forbidden。200 OK 代表 B 同意進行語音通訊，403 Forbidden 則代表拒絕進行語音通訊。

若 A 在10秒內沒有收到 B 回傳的 Data，則會判斷該 Interest 已經 Timeout 了，代表 B 沒有選擇接受或拒絕。

doi:10.6837/ncnu201900270

值得一提的是，稍早在 Session Module 的 486 Busy 是用於表示對方已經在別的會話之中，而 VoiceChat Module 則是雙方已在同一會話的情況下，邀請對方增加聲音資料交換的功能。因此，這裡不會出現 486 Busy 的訊息。

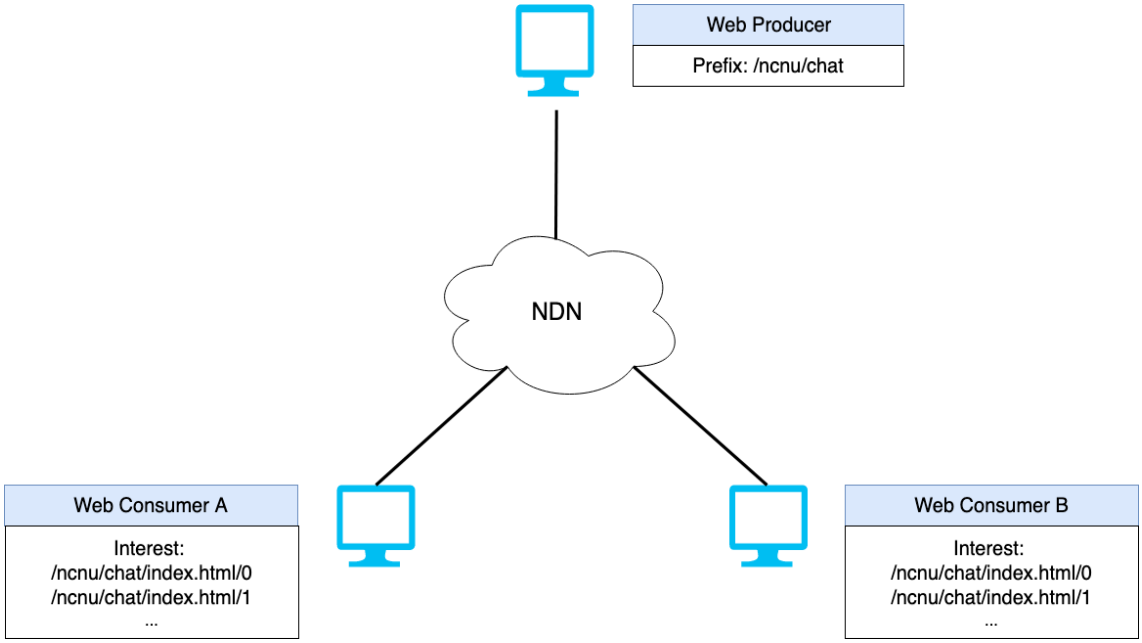
VoiceChat Module 的聲音資料 (Audio Data) 是透過 RecordRTC [21] 將聲音錄製下來，並且雙方持續地向對方要求當前所錄製的聲音資料並且播放，進而達到語音通訊的功能，如圖四十六所示。A 向 B 要求聲音資料，會使用 Interest，Name 為 /ncnu/chat/B/VoiceChat/AudioData。B 則會回應 Data，Name 為 /ncnu/chat/B/VoiceChat/AudioData，並在 Content 放入聲音的資料內容。

doi:10.6837/ncnu201900270

第四章 系統實作

前章已詳細介紹本系統的架構與應用的運作流程，接下來說明系統實作的細節。

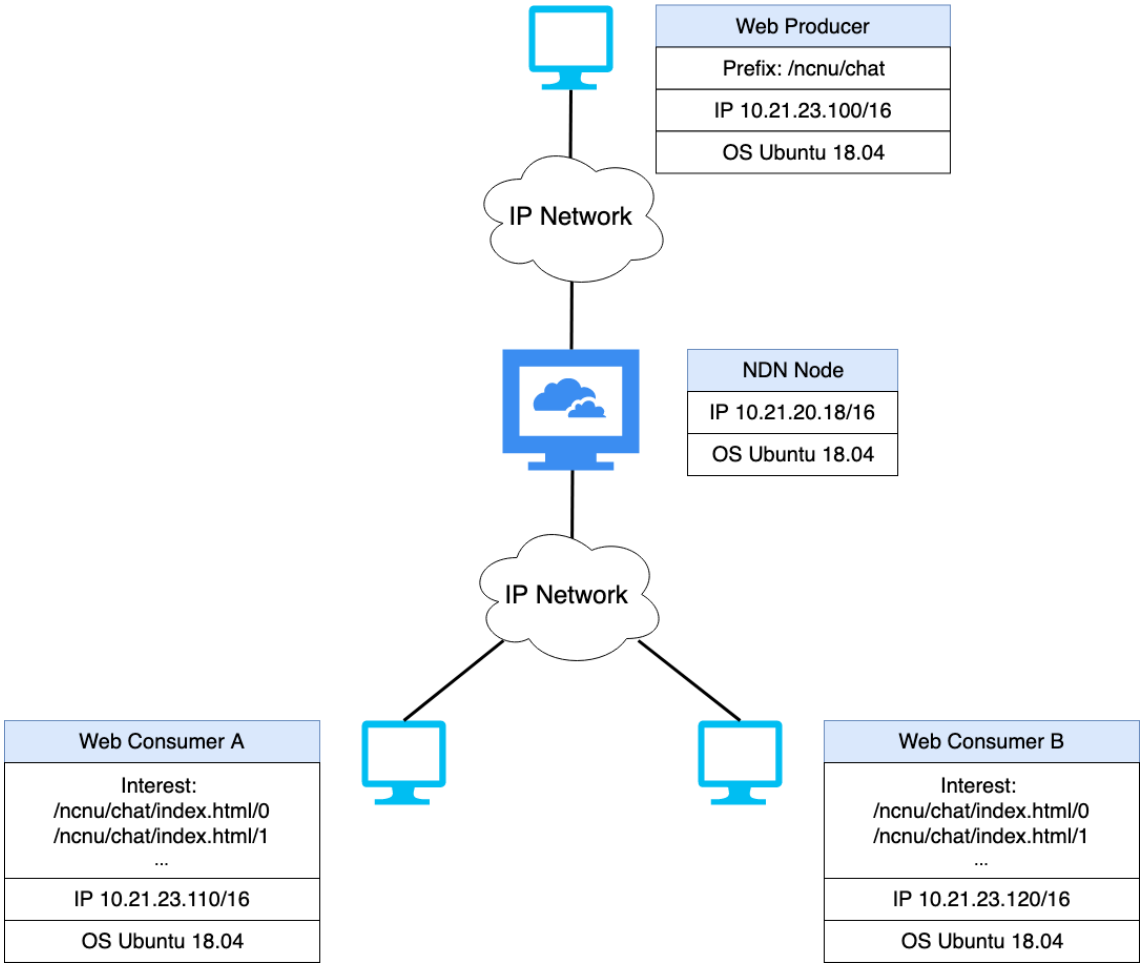
第一節 網路架構



圖四十七、NDN 網路架構

理想的狀況下，NDN 中的所有裝置，不需要靠額外的幫助即可完成與其他裝置的溝通。如圖四十七所示，Web Producer 可以直接在 NDN 之中註冊 Prefix /ncnu/chat，而 Web Consumer 也只要將 Interest 送出，NDN 便會根據匹配的 Prefix，將 Interest 送到 Web Producer，進而取得 Data 的回應。

doi:10.6837/ncnu201900270



圖四十八、網路架構

然而，目前 NDN 在發展之初尚未提供 native mode，因此實驗的部分，仍是以 Tunnel 的方式運作在 IP 網路之上。本實驗中每台機器皆使用 Ubuntu 18.04 作業系統，並且配置了一個 IP 位址，如圖四十八所示。透過與 NDN Node 建立 Tunnel 的方式，進行彼此間 NDN 封包的傳輸。

第二節 NDN Node

NDN Node 使用 NDN Forwarding Daemon (NFD) 提供 NDN 的網路服務，如：建立 Tunnel、註冊 Prefix 以及透過 Tunnel 進行 NDN 封包的傳遞。

Web Producer、Web Consumer 跟 NDN 網頁即時通訊應用都會與 NDN Node 建立 Tunnel。其中，Web Producer 與 Web Consumer 會與 NDN Node 的 Port: 6363 建立 TCP 連線，並以此 TCP 連線作為 Tunnel；NDN 網頁即時通訊應會與

doi:10.6837/ncnu201900270

NDN Node 的 Port: 9696 建立 WebSocket 連線，並以此 WebSocket 連線作為 Tunnel。

第一小節 WireShark 新增對 NDN 封包的解析

>> git clone https://github.com/named-data/ndn-tools.git
>> sudo cp ndn-tools/tools/dissect-wireshark/ndn.lua\
/usr/share/wireshark/ndn.lua

支援 Wireshark 對 NDN 封包的分析，可以使用 [22] 所提供的 dissect-wireshark，將 tools/dissect-wireshark 內的 ndn.lua 放至 Wireshark 的安裝目錄 (與 init.lua 同目錄)，Ubuntu 18.04 安裝 Wireshark 請參考附錄一，預設安裝目錄為 /usr/share/wireshark。

```
-- deprecated function names
datafile_path = Dir.global_config_path
persconffile_path = Dir.personal_config_path


dofile(DATA_DIR.."console.lua")
dofile(DATA_DIR.."ndn.lua")
--dofile(DATA_DIR.."dtd_gen.lua")
~
~
```

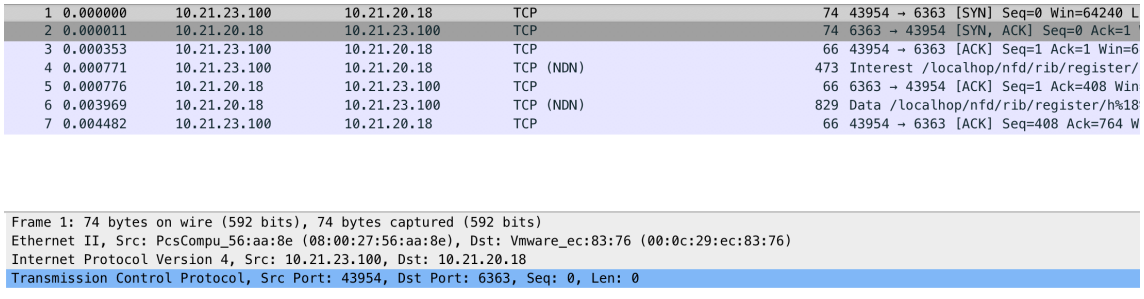
圖四十九、Wireshark 新增 NDN 封包解析 (init.lua)

>> sudo vim /usr/share/wireshark/init.lua

修改 Wireshark 安裝目錄下的 init.lua，在 dofile(DATA_DIR.. "ndn.lua") 下方，新增 dofile(DATA_DIR.. "ndn.lua")，如圖四十九所示。

第二小節 TCP Tunnel 的建立

doi:10.6837/ncnu201900270



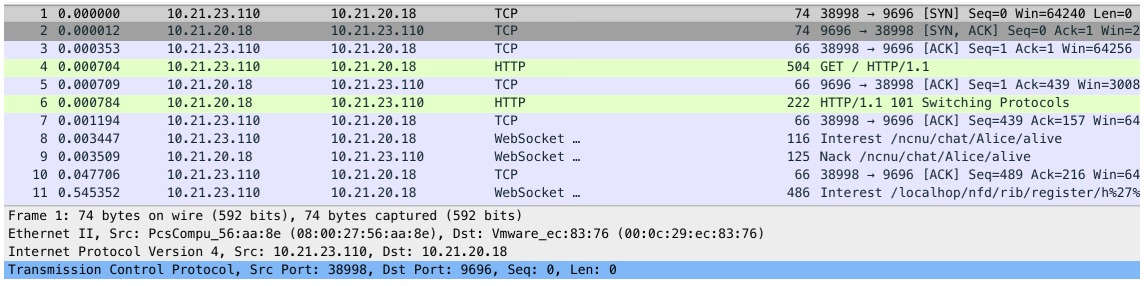
圖五十、建立 Tunnel (TCP)

Web Producer 與 Web Consumer 是與 NDN Node 的 Port: 6363 進行 TCP 連線，並以此 TCP 連線作為 Tunnel，傳遞 NDN 的封包。

圖五十為 Web Prodcuer 建立 Tunnel 的過程，封包編號 1~3 顯示，Web Producer 會先與 NDN Node 進行 TCP 的三方交握。完成 TCP 連線的建立之後，Web Producer 會以此 TCP 連線作為 Tunnel。看到封包編號 4~7，Web Producer 透過此 Tunnel (TCP 連線)，達到 NDN 封包的傳遞。

封包資料是在 NDN Node 上抓取，網路卡為 ens160；網路卡的 IP Address 為 10.21.20.18；Capture Filter 為 ip host 10.21.23.100 (Web Producer 的 IP 位址)。用於觀察 Web Producer 的 Tunnel 建立，以及建立之後，NDN 封包透過 Tunnel 的傳遞過程。

第三小節 WebSocket Tunnel 建立



圖五十一、建立 Tunnel (WebSocket)

NDN 網頁即時通訊應用是與 NDN Node 的 Port: 9696 進行 WebSocket 連線，並以此 WebSocket 連線作為 Tunnel，傳遞 NDN 的封包。

圖五十一為 NDN 網頁即時通訊應用建立 Tunnel 的過程，封包編號 1~3 顯示，NDN 網頁即時通訊應用會先與 NDN Node 進行 TCP 的三方交握，完成 TCP 連線建立之後，封包編號 4~7 顯示，NDN 網頁即時通訊應用申請將此 TCP 連線

doi:10.6837/ncnu201900270

升級成 WebSocket。完成 WebSocket 的建立之後，NDN 網頁即時通訊應用會將此 WebSocket 作為 Tunnel。可以看到封包編號 8~11，NDN 網頁即時通訊應用透過此 Tunnel (WebSocket 連線)，達到 NDN 封包的傳遞。

NDN 網頁即時通訊應用的封包資料是在 NDN Node 上抓取，網路卡為 ens160；IP Address 為 10.21.20.18；Capture Filter 為 ip host 10.21.23.110 (NDN 網頁即時通訊應用的使用者 IP 位址)。用於觀察 NDN 網頁即時通訊應用的 Tunnel 建立，以及建立之後，NDN 封包透過 Tunnel 的傳遞過程。

第四小節 NFD 安裝

1. sudo apt-get install software-properties-common
2. sudo add-apt-repository ppa:named-data/ppa
3. sudo apt-get update
4. sudo apt-get install nfd

第五小節 允許 Tunnel 建立

1. sudo vim /etc/ndn/nfd.conf
2. Uncomment localhop_security， line 338 ~ 344（圖四十八）

```
localhop_security
{
  trust-anchor
  {
    type any
  }
}
```

圖五十二、取消 localhop_security 註解

第六小節 重啟 NFD

1. sudo nfd-stop
2. sudo nfd-start

第三節 Web Producer

doi:10.6837/ncnu201900270

Web Producer 負責提供 NDN 網頁存取服務，使用 NDN 的 Python 套件為 PyNDN [23]，透過其所提供的 library，Web Producer 可以完成以下幾種功能。

1. 指定 NDN Node 的 IP 位址

   face = Face("10.21.20.18")

   新增物件 Face 並且指定 NDN Node 的 IP 位址。

2. 建立 Tunnel 與註冊 Prefix

   face.registerPrefix 會先完成 Tunnel 的建立，接著才能向 NDN Node 註冊 Prefix 以及傳遞 NDN 封包。

   透過 face.registerPrefix，Web Producer 在進行註冊之前，會先向 NDN Node 發起 TCP 連線的建立。完成 TCP 連線的建立之後，Web Producer 會以此 TCP 連線作為 Tunnel，向 NDN Node 送出註冊 Prefix 的訊息。

3. 回應 Data

   face.registerPrefix(prefix, onInterest, onRegisterFailed) 為註冊 Prefix 的 function。第一個參數為註冊的 Prefix；第二個參數 onInterest 為收到 Interest 後，回應 Data 的 function；第三個參數 onRegisterFailed 為註冊失敗時，執行的 function。

第一小節 安裝 PyNDN

1. sudo apt-get install python3 python3-pip # Python3 Version = 3.6.7
2. pip3 install pyndn # PyNDN Version = 2.10b1

第二小節 設定 Web Producer

```
1 [NDN Node]
2 IPAddress = 10.21.20.18
```

圖五十三、設置 NDN Node 的 IP 位址 (webProducer.conf)

>> vim webProducer.conf

doi:10.6837/ncnu201900270

完成 PyNDN 的安裝之後，會需要設定 NDN Node 的 IP Address，此時只要修改 Web Producer 目錄下的 webProducer.conf 即可，將 [NDN Node] 的 IPAddress 設置為 NDN Node 的 IP 位址，如圖五十三。

```
1 const ndnNodeIPAddress = "10.21.20.18"
```

圖五十四、設置 NDN Node 的 IP 位址 (www/js/main.js)

>> vim www/js/main.js

除了 Web Producer 之外，還需設定 NDN 網頁即時通訊應用的 NDN Node。如圖五十四所示，修改 Web Producer 目錄下的 www/js/main.js，將 ndnNodeIPAddress 設置為 NDN Node 的 IP Address。

第三小節 Web Producer 實作

```
lab409@webproducer:~/Documents/WebProducer$ python3 webProducer.py
Register prefix /ncnu/chat
```

圖五十五、執行 Web Producer

>> python3 webProducer.py

設置好 NDN Node 的 IP 位址之後，接著在 WebProducer 的目錄下執行 webProducer.py，如圖五十五。

Web Producer 使用 PyNDN 中的 face.registerPrefix 註冊 Prefix (/ncnu/chat)，當收到 Interest ，會取 Prefix 之後的 Name，如：/ncnu/chat/index.html/0，取 index.html，0 為序列號碼。以此得知 Web Consumer 所要求的資料為何，接著使用 Python 的 open 將檔案以位元的形式讀入，f = open (filePath, "rb")。取得資料後，根據序列號碼，決定回傳 Data 的 Content 內容。最後，呼叫 face.putData(data)，將 Data 送出。

doi:10.6837/ncnu201900270

在 NDN 中，根據 PyNDN 提供的 Library，其最大傳輸單元 (Maximum Transmission Unit - MTU) 為 8800 bytes，而檔案本身的大小可能超過此限制，因此，會在 Name 的後方增加序列號碼
(如：/ncnu/chat/index.html/0、/ncnu/chat/index.html/1)
，將較大的檔案分割成較小的檔案，並且以空字串代表檔案已傳輸完畢。

第四節 Web Consumer

前人的 NDN Browser 是建立在 C++ 之上的，本論文中的 Web Consumer 參考其底層傳輸及網頁的要求與呈現方式，並且改用 PyQt5 與 PyNDN 進行實作。PyQt5 為一款 Python 的圖形使用者介面程式，在網頁的部分，使用的是 PyQt5 其中的一個套件 QWebEngine [24]，核心為 Google 從 Webkit 分支 blink 上，所開發的 Chromium，其功能相當齊全。搭配 PyNDN 所提供的函式庫，即可完成以 Python 執行，具備要求網頁功能的 Web Consumer。

第一小節 安裝 PyQt5 與 PyNDN

1. sudo apt-get install python3 python3-pip # Python3 Version = 3.6.7
2. pip3 install pyqt5==5.11.2 # PyQt5 Version = 5.11.2
3. pip3 install pyndn # PyNDN Version = 2.10b1

   (本實驗所使用的作業系統為 Ubuntu 18.04 LTS，經過測試，PyQt5 在 5.11.1 之前的版本都會產生問題，因此在這一小節所附註的版本是推薦使用的，能成功地進行實作。)

第二小節 PyQt5 與 PyNDN 函式介紹

PyNDN:

Face:

face = Face ("10.21.20.18")

新增物件 Face 並且指定 NDN Node 的 IP 位址。

face.expressInterest:

face.expressInterest 會先完成 Tunnel 的建立，接著便能在此 Tunnel

doi:10.6837/ncnu201900270

中，傳遞 NDN 封包。透過此 face.expressInterest，Web Consumer 在送出 Interest 之前，會先向 NDN Node 建立 TCP 連線，接著以此 TCP 連線作為 Tunnel，向 NDN Node 送出 Interest 及接收回應的 Data。

face.expressInterest (interest, onData, onTimeout):

第一個參數想要送出的 Interest；第二個參數 onData 為收到 Data 後所執行的 function；第三個參數 onTimeout 為 Interest 已經 Timeout 後，所執行的 function。

PyQt5:

BrowserWindow:

建立應用程式的使用介面。

PyQt5 QWebEngine:

QWebEngineView:

用於顯示網頁。

QWebEngineUrlSchemeHandler:

用於自訂的協定，其標籤來源以何種方式進行處理，並且回傳資料內容給 QWebEngineView。

QIODevice

用於將資料的讀寫。在此處會與 QWebEngineSchemeHandler 搭配，當 QWebEngineSchemeHandler 取得資料內容後，再透過 QIODevice 將其讀入。

第三小節 設定 WebConsumer

```
1 [NDN Node]
2 IPAddress = 10.21.20.18
```

圖五十六、設置 NDN Node 的 IP 位址 (webConsumer.conf)

doi:10.6837/ncnu201900270

>> vim webConsumer.conf

完成 PyQt5 與 PyNDN 的安裝之後，會需要設定 NDN Node 的 IP Address，此時只要修改 Web Consumer 目錄下的 webConsumer.conf 即可，將 [NDN Node] 的 IPAddress 設置為 NDN Node 的 IP 位址，如圖五十六。

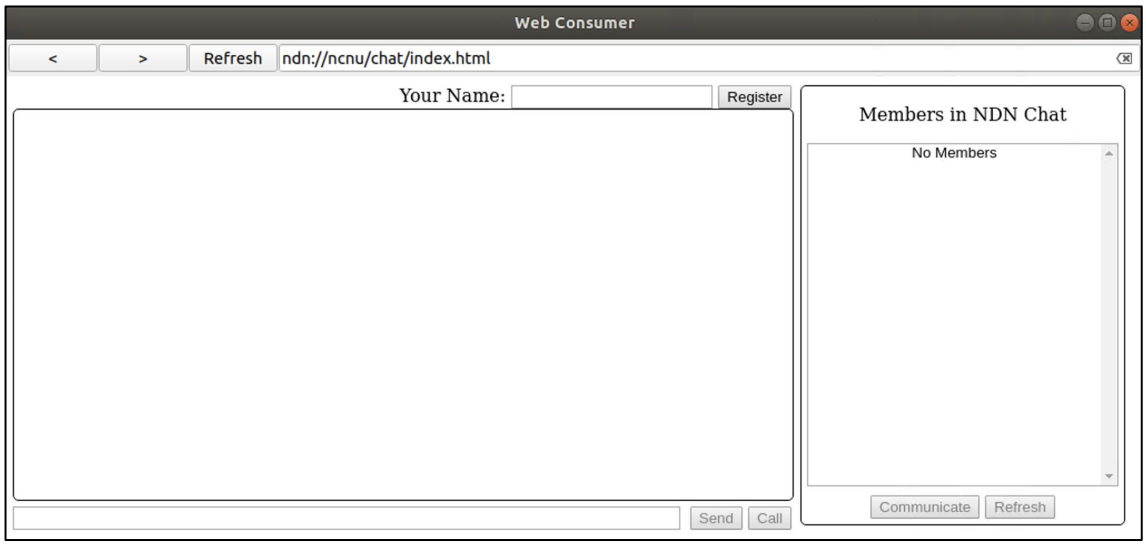
第四小節 Web Consumer 實作

```
lab409@webconsumer:~/Documents/WebConsumer$ python3 main.py
```

圖五十七、執行 Web Consumer

>> python3 main.py

設置好 NDN Node 的 IP 位址之後，接著在 WebConsumer 的目錄下執行 main.py，如圖五十七。



圖五十八、NDN 網頁即時通訊應用畫面

成功執行 Web Consumer 之後，只要在上方的 URL 欄位輸入，ndn://ncnu/chat/index.html，即可向 Web Producer 進行 NDN 網頁即時通訊應用的要求，取得之後的畫面如圖五十八所示。

doi:10.6837/ncnu201900270

為了建立 Web Consumer，主要的重點在於，NDN 協定的 URL 輸入；以及解析 HTML 時，標籤的來源亦可支援 NDN 協定的 URL 輸入。

首先，Web Consumer 的 Browser UI 是利用 BrowserWindow 所建立而成的，並且提供一個可以輸入的 URL 列。

在收到以 ndn:// 為開頭的 URL 後，會將其作為 Interest 的 Name ，
如：ndn://ncnu/chat/index.html -> Interest /ncnu/chat/index.html
並且在 Name 的後方加上序列號碼，
如：/ncnu/chat/index.html/0、/ncnu/chat/index.html/1
，然後透過 NDN Module 將其送出。

這部分使用到 PyNDN 中的 face.expressInterest，在送出 Interest 之前，會先與 NDN Node 建立起 TCP 連線，並以此 TCP 連線作為 Tunnel，達到 Interest 與 Data 的收送。

隨著序列號碼持續增加，當收到 Data 的 Content 內容為空字串時，即可得知此檔案已傳輸完畢。由於會將 HTML 檔案會被切成多個小片段傳遞，因此，Web Consumer 會先將其重組，完成重組後才會讀入。

此時產生了一個問題。剛剛輸入的 URL 為 NDN 協定 (ndn://)，然而，在讀入 HTML 後，由於負責網頁畫面輸出的 QWebEngineView 不認識 NDN 協定，造成 QWebEngineView 會以 Text 形式輸出此 HTML 檔案的內容。要解決不支援 NDN 協定的問題，需要在 QWebEngine 已知的協定中 (如: http、https、file)，新增支援 ndn。

不過 QWebEngine 目前在新增協定的方法上並不完備(缺少函式)，於是，暫時性的解決方法是先將 HTML 儲存為暫存檔案，以 file 的形式讀入，如此即可完成 HTML 的網頁輸出。

取得網頁並且成功輸出畫面後，接著，HTML 檔案中亦有以 URL 做為資料來源使用的標籤，如: <img>、<audio>、<video>、<javascript> 等。因此，除了從網址列輸入的 URL 要能支援 NDN 協定之外，標籤的資料來源同樣也需要支援 NDN 協定的 URL 輸入。而不是使用 NDN 取得網頁後，卻只能以 http 或者 https 等其他協定要求標籤檔案。

doi:10.6837/ncnu201900270

在解析 HTML 檔案的時候，瀏覽器會使用 QWebEngineUrlSchemeHandler 特別處理資料來源為 URL 的標籤。QWebEngineUrlSchemeHandler 會取得這些資料來源的 URL，並分析該 URL (如：ndn://ncnu/chat/js/main.js) 所要求的內容，接著，作為 Interest 的 Name，並且在 Name 的最後方加上序列號碼，
如：Interest /ncnu/chat/js/main.js/0、/ncnu/chat/js/main.js/1
，然後將其送出。

此處同樣是使用 PyNDN 的 face.expressInterest，在已建立的 Tunnel 上，進行 NDN 封包的傳遞。隨著序列號碼持續增加，當收到 Data，其 Content 內容為空字串，即可得知此檔案已傳輸完畢。這就是 HTML Parse and Render Module 所完成的部分。

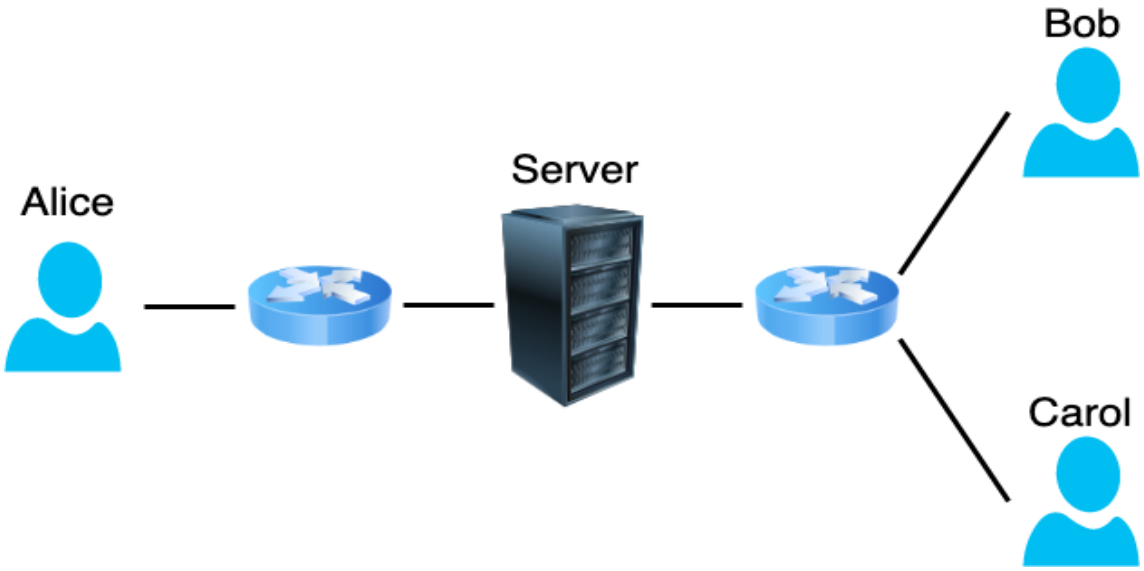
完成檔案的傳輸之後，QWebEngineSchemeHandler 會使用 QIODevice 將檔案內容讀入，並且將其內容交給 QWebEngineView 建構輸出畫面。在收齊所有資料後，才會完成整個網頁的建構，最後由 QWebEngineView ，也就是 Browser UI 顯示網頁內容。

第五節 網頁即按即說應用 (Multicast)

網頁即時通訊應用雖然方便，但一對一的通訊還不足以展現出 Multicast 的效果。參考了 [25] 之後，我們發現即按即說應用（Push-to-Talk）非常適合用來展現 NDN 網路中 Multicast 機制的威力。即按即說應用是指，在同一群組內的所有使用者，每人都擁有一具備按鈕的裝置 (手機 App 或是無線電對講機等)。如果 Alice 有訊息要傳遞給群組內的人知道的話，Alice 會按下按鈕，取得群組內的發話權。此時，群組內的其他人會無法按下按鈕，只能單方向接收到 Alice 的語音訊息。直到 Alice 傳遞完語音訊息及釋出發話權 (放開按鈕)，其他人才能按下按鈕，取得發話權。

doi:10.6837/ncnu201900270

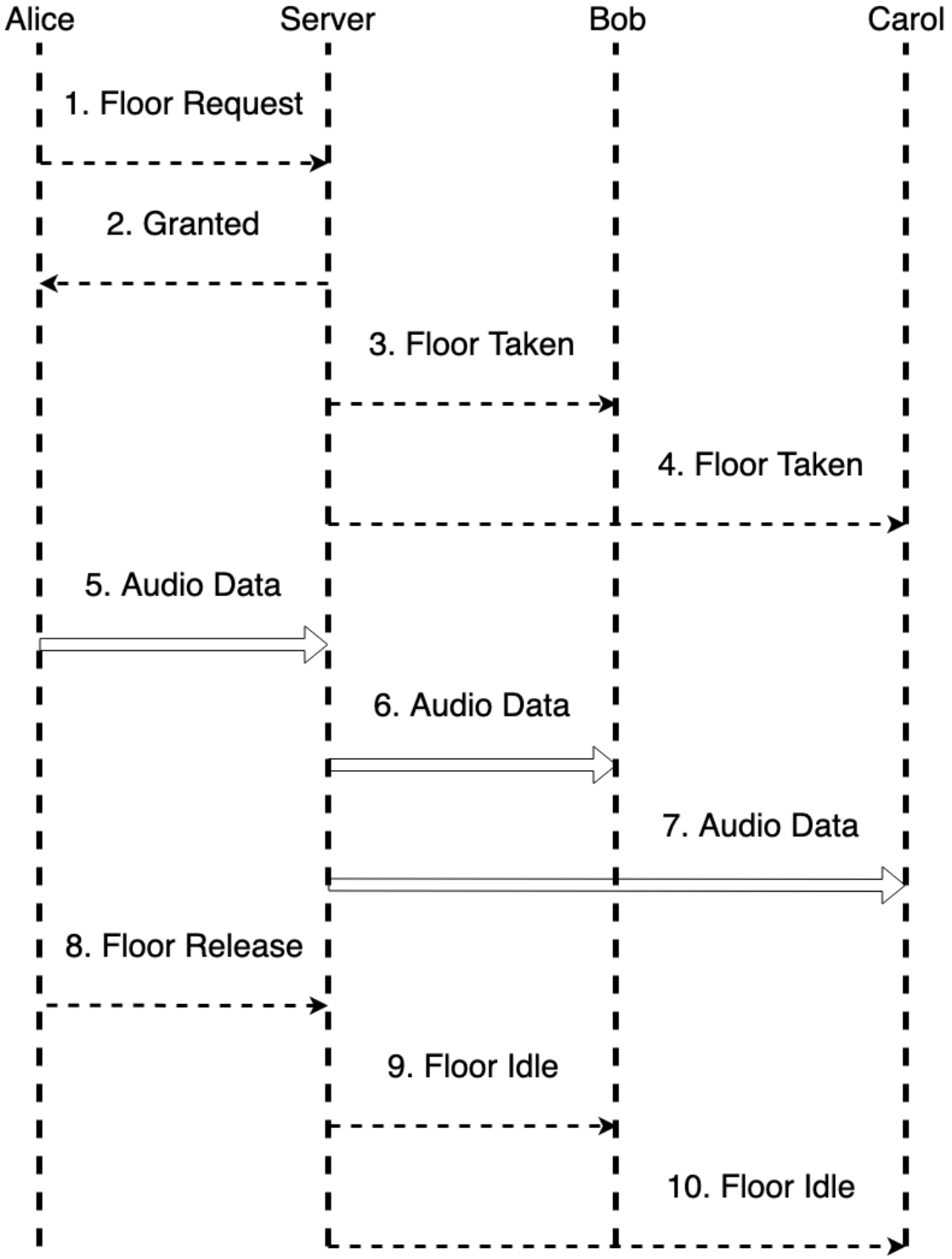
第一小節 IP 網路的即按即說應用



圖五十九、即按即說應用

IP 網路中，即按即說應用是利用 Server，建立使用者之間的關聯 (群組) 與轉送彼此的訊息[26]，如：取得發話權、傳遞語音資料以及釋出發話權等，其架構如圖五十九所示。

doi:10.6837/ncnu201900270



圖六十、即按即說應用操作流程

圖六十為即按即說應用的操作流程。假設，Alice、Bob、Carol 在同一群組，並且 Alice 有語音訊息要傳遞給同群組的其他人 (Bob、Carol)。Alice 會按下按鈕，向 Server 要求發話權 (Floor Request)，接著 Server 會允許 Alice 的要求 (Granted)，

doi:10.6837/ncnu201900270

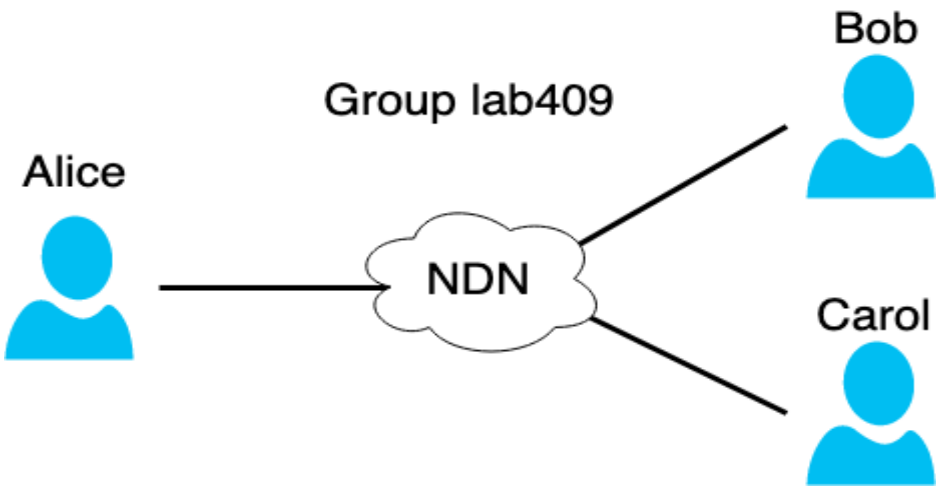
並且向 Bob 與 Carol 告知，發話權已被 Alice 拿走了 (Floor Taken)。

取得發話權之後的 Alice 便會開始發送語音訊息給 Server，由 Server 將語音訊息轉送至同群組內的 Bob 與 Carol，此時的 Bob 與 Carol 會無法取得發話權，只能收到 Alice 的語音訊息。

若是要取得發話權，必須等到 Alice 發話完畢且向 Server 送出釋出發話權的訊息 (Floor Release) 後，Bob 與 Carol 才會從 Server 收到目前發話權已經釋出的訊息 (Floor Idle)。接著，才能按下按鈕取得發話權。

第二小節 NDN 即按即說應用

上面簡單回顧了 IP 網路的即按即說應用是如何操作的，接下來會說明，本研究是如何利用 NDN Multicast 機制，達到 NDN 即按即說應用的功能



圖六十一、NDN 網頁即按即說應用

在 NDN 中，透過命名資料的方式，確認使用者是否存在、搜尋使用者以及加入群組，這些功能與網頁即時通訊應用的模組相同，此處將不贅述。

NDN 網頁即按即說應用會展現，使用者如何取得發話權、傳遞語音資料以及釋出發話權等，其架構如圖六十一所示。

doi:10.6837/ncnu201900270

```
Check User Alive                                              main.js:303
Interest: /ncnu/ptt/user/Carol/alive is Nack                  main.js:511
/ndn/broadcast/ncnu/ptt/user is registered                    main.js:158
/ncnu/ptt/user/Carol is registered                            main.js:159
User Searching Thread is started                              main.js:324
Group Searching Thread is started                             main.js:333
Add User: Alice                                               main.js:424
Add Group: lab409                                             main.js:432
Add User: Bob                                                 main.js:424
Selected group name: lab409                                    main.js:95
/ndn/broadcast/ncnu/ptt/group is registered                   main.js:173
/ncnu/ptt/group/lab409/alive is registered                    main.js:174
/ncnu/ptt/group/lab409/user/Carol is registered               main.js:175
/ncnu/ptt/group/lab409/list is registered                     main.js:176
/ncnu/ptt/group/lab409/list/add/Carol                         main.js:345
▶ (3) ["Alice", "Bob", "Carol"]                               main.js:460
```

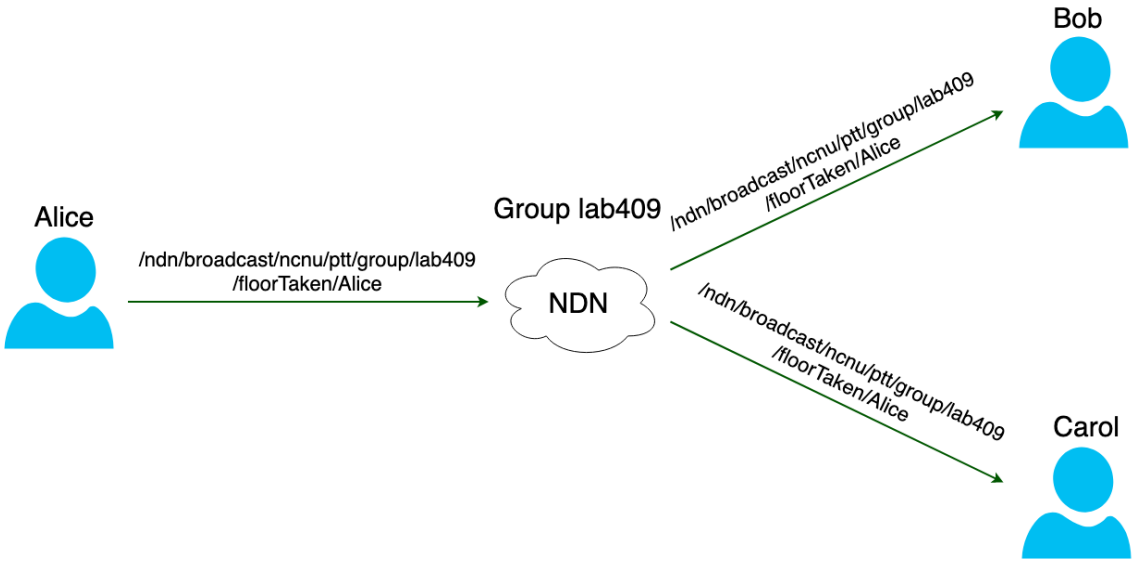
圖六十二、群組使用者名單

圖六十三、NDN 網頁即按即說應用畫面

在 NDN 網頁即按即說應用中，每一位使用者在加入群組後，都會得到該群組的名單，如圖六十二所示。接著，進入群組內的操作介面，提供了一個發話

doi:10.6837/ncnu201900270

按鈕，讓使用者可以取得群組的發話權。同時，在按鈕上方，會根據名單資訊，顯示當前的群組人數，如圖六十三所示。



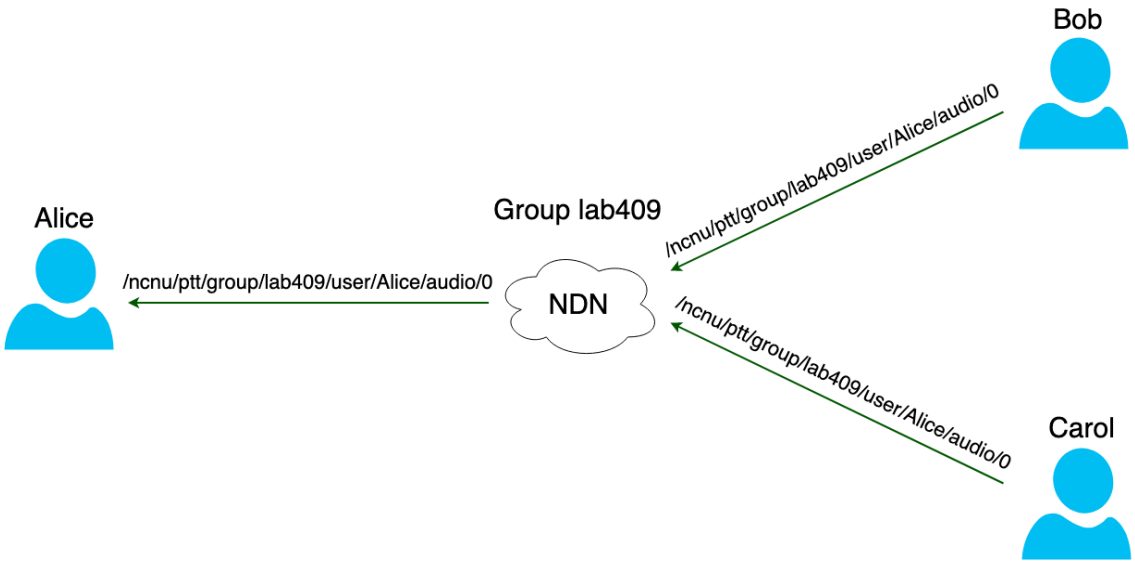
圖六十四、NDN 要求發話權

假設 Alice 、Bob 與 Carol 都已加入群組 lab409 ，且所有人都註冊了 Prefix /ndn/broadcast/ncnu/ptt/group/lab409，其中，/ndn/broadcast 為 NDN 的廣播模式。

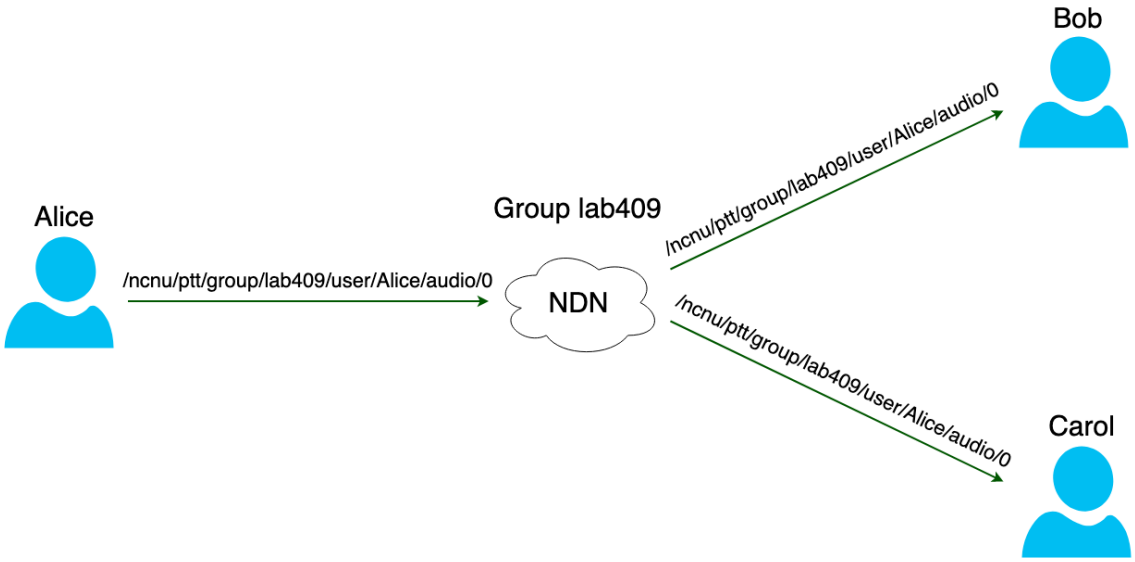
當 Alice 有語音訊息要傳遞給群組內其他人時，Alice 會按下按鈕並且發送 Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice，此 Interest 的 Name 匹配 Prefix /ndn/broadcast/ncnu/ptt/group/lab409。加上，NDN 會根據 /ndn/broadcast，判斷此 Interest 是廣播封包，因此會將 Interest 送至所有註冊了 Prefix /ndn/broadcast/ncnu/ptt/group/lab409 的使用者。

接著，群組內的 Bob 與 Carol 便會收到 Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice，得知目前群組 lab409 的發話權已經被 Alice 取走 (Floor Taken)，如圖六十四所示。

doi:10.6837/ncnu201900270



圖六十五、NDN 要求語音訊息



圖六十六、NDN 回應語音訊息

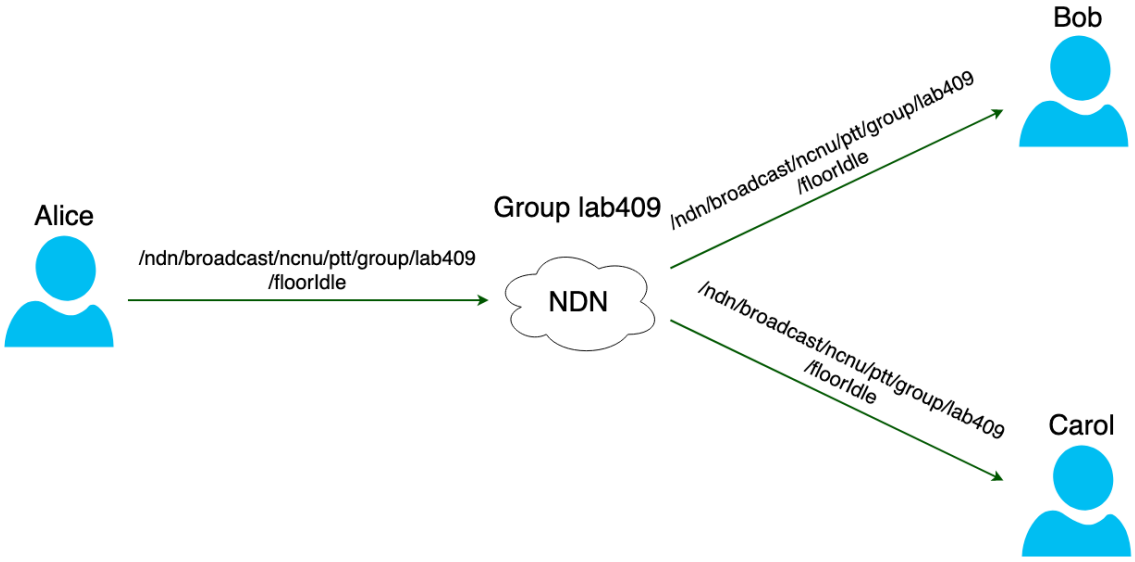
此時，Bob 與 Carol 將無法取得發話權，並且會向 Alice 要求語音訊息，/ncnu/ptt/group/lab409/user/Alice/audio/0 (0 為序列號碼)。在這階段，會使用到 NDN Multicast 機制，當 Bob 與 Carol 把相同 Name 的 Interest 送到 NDN 的時候，由於要求的是同一份資料，因此，NDN 會將 Bob 與 Carol 的 Interest 及其進入的介面記錄在 Pending Interest Table 內，然後，合併成一份 Interest 送到 Alice，如圖六十五所示。

Alice 發出的語音訊息，會以 Data 封包回應，NDN 會根據 Pending Interest Table 的記錄，把 Data 複製成多份，分別傳遞給 Bob 與 Carol，以達到 NDN

doi:10.6837/ncnu201900270

Multicast 機制，如圖六十六所示。

在 IP 網路中，發話者是將語音資料送往 Server，再由 Server 將資料傳遞至同群組下的使用者。相較之下，因為 NDN 的機制，發話者在送出語音資料時，NDN 能夠自動將 Data 複製成多份，並傳遞給其他使用者，充分地展現出 NDN Multicast 的特色，無疑是較有效率的做法。



圖六十七、NDN 釋出發話權

最後，當 Alice 結束發話的時候，會送出 /ndn/broadcast/ncnu/ptt/group/lab409/floorIdle，此 Interest 的 Name 匹配 Prefix /ndn/broadcast/ncnu/ptt/group/lab409。加上，NDN 會根據 /ndn/broadcast，判斷此 Interest 是廣播封包，因此會將 Interest 送至所有註冊了 Prefix /ndn/broadcast/ncnu/ptt/group/lab409 的使用者。

接著，群組內的 Bob 與 Carol 便能得知目前發話權已經釋出 (Floor Idle)，如圖六十七所示。此時，群組內的所有人都可按下按鈕取得發話權，發送語音訊息。

doi:10.6837/ncnu201900270

第五章 實驗結果

本研究提出 NDN 的網頁即時通訊應用，經過實際部署與測試。可以透過 Web Consumer A 與 Web Consumer B 向 Web Producer 取得應用，在 NDN 的網路節點上觀察到快取機制。

實驗中各個設備的 IP 位址如下：

1. NDN Node – 10.21.20.18
2. Web Producer – 10.21.23.100
3. Web Consumer A – 10.21.23.110
4. Web Consumer B – 10.21.23.120

在 NDN Node 上，使用 Wireshark 進行觀察。

NDN Node:

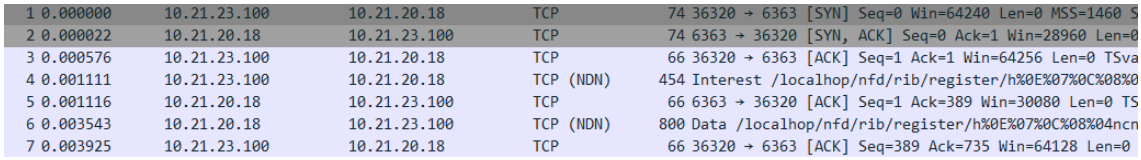
網路卡: ens160

IP Address: 10.21.20.18

Capture Filter: tcp port 6363

(監聽 Tunnel 的封包)

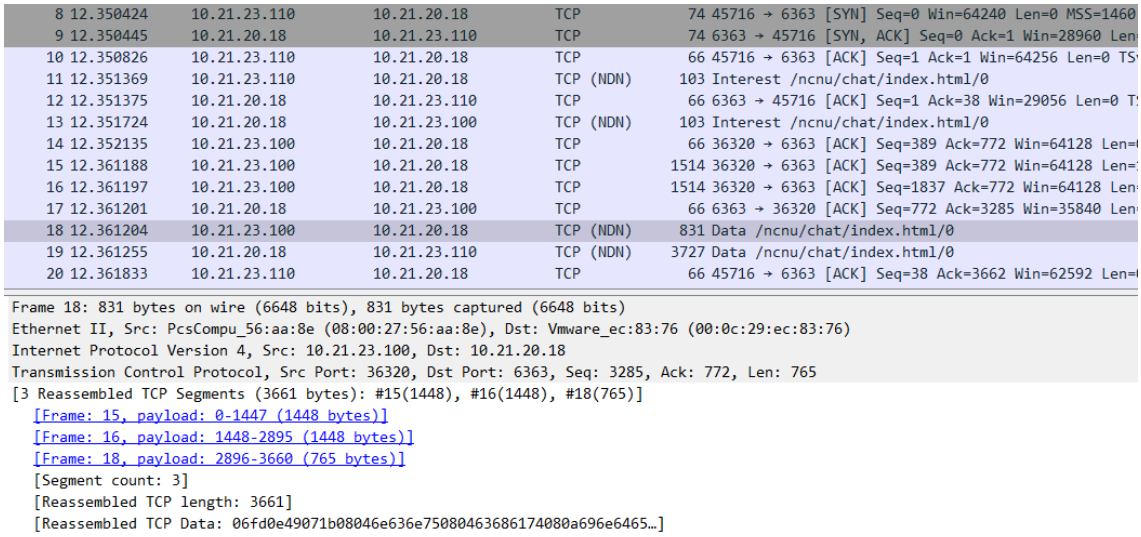
(如未新增 Wireshark 對 NDN 封包的解析，請參考 第4.2.1小節)



圖六十八、TCP 連線建立

Web Producer 與 Web Consumer 會在 TCP Port 6363 上建立的 TCP 連線，並以此為 Tunnel，傳遞 NDN 封包。如圖六十八所示，Web Producer 與 NDN Node 建立 TCP 連線 (封包編號 1~3)，並以此連線作為 Tunnel，向 NDN Node 註冊 Prefix 的封包資料 (封包編號 4~7)。其中，Interest 或者 Data 封包是以帶有資料內容的 TCP 封包 [PSH、ACK] 推送出去的，對方在收到之後會回傳 TCP ACK 封包 (如封包編號 4~5 以及 封包編號 6~7 所示)。

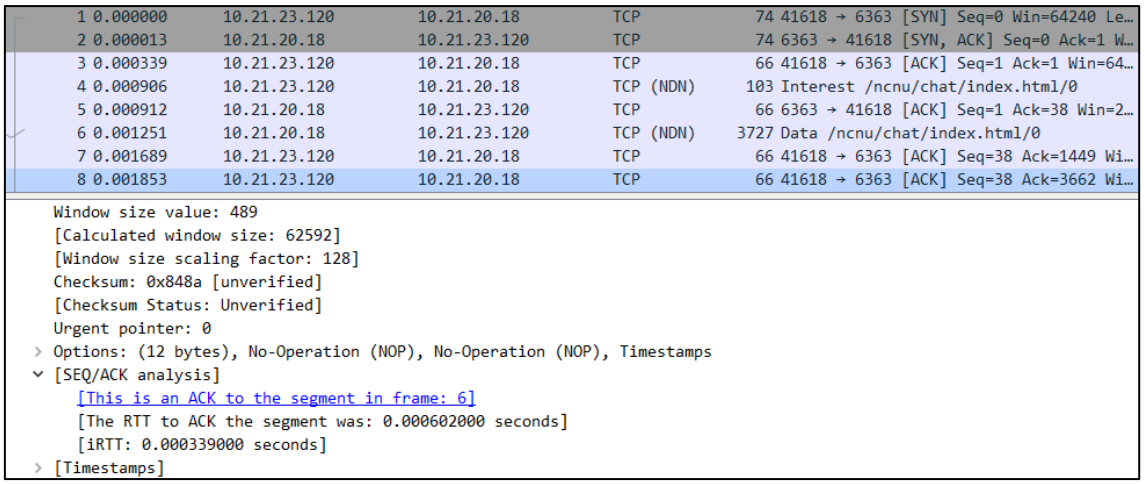
doi:10.6837/ncnu201900270



圖六十九、第一次要求網頁

如圖六十九所示，Web Consumer A 進行第一次網頁要求的封包資料。Web Consumer A 會先與 NDN Node 建立 TCP 連線 (封包編號 8~10)，並以此連線作為 Tunnel，收送 NDN 封包。

Web Consumer A (10.21.20.110) 會送出要求 NDN 網頁即時通訊應用的 Interest (封包編號 11~12)，NDN Node (10.21.20.18) 在收到之後，因為並無快取過，因此會送往 Web Producer (封包編號 13~14)。接著，Web Producer (10.21.20.100) 會回應 Data (封包編號 15~18，封包編號 18 為 15、16、18 重組而成)，再由 NDN Node 將 Data 傳遞給 Web Consumer A (封包編號 19~20)。

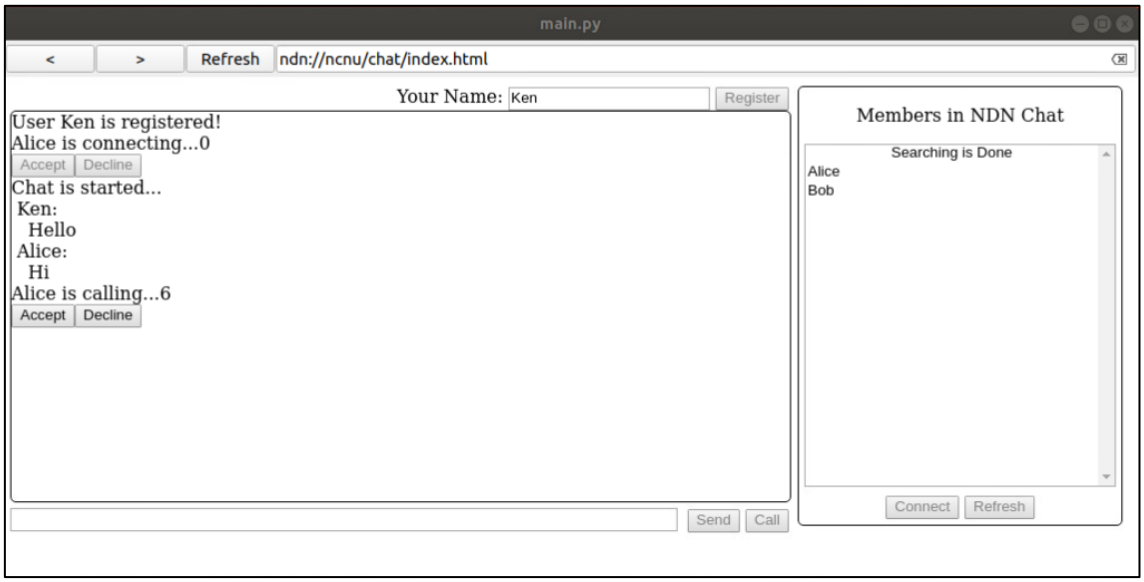


圖七十、第二次要求網頁

doi:10.6837/ncnu201900270

如圖七十為 Web Consumer B 第二次網頁要求的封包資料。Web Consumer B 會先與 NDN Node 建立 TCP 連線 (封包編號 1~3)，並以此連線作為 Tunnel，收送 NDN 封包。

Web Consumer B (10.21.20.120) 會送出要求 NDN 網頁即時通訊應用的 Interest (封包編號 4~5)，NDN Node (10.21.20.18) 在收到之後，因為已經快取過了，因此不會將 Interest 送往 Web Producer (10.21.20.100)，而是直接回應 Data 給 Web Consumer B (封包編號 6~8，封包編號 6 為重組過的封包，封包編號 7~8 為重組前的 ACK 封包)



圖七十一、NDN 網頁即時通訊應用操作

在取得網頁即時通訊應用之後，不管是在註冊、搜尋其他使用者還是訊息交換方面，都跟 IP 網路上的聊天室使用方法相近，如圖七十一所示。

此外，本研究亦將一對一的網頁即時通訊，進一步擴展為網頁即按即說的應用，可以用於觀察 NDN Multicast 的功能。

實驗中各個設備的 IP 位址如下：

1. NDN Node – 10.21.20.18
2. Alice (Talker) – 10.21.20.83
3. Bob (Listener) – 10.21.23.110

doi:10.6837/ncnu201900270

4. Carol (Listener) – 10.21.23.120

在 NDN Node 上，使用 Wireshark 進行觀察。

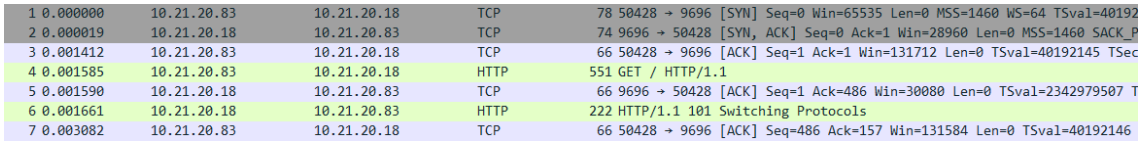
NDN Node:

網路卡: ens160

IP Address: 10.21.20.18

Capture Filter: tcp port 9696

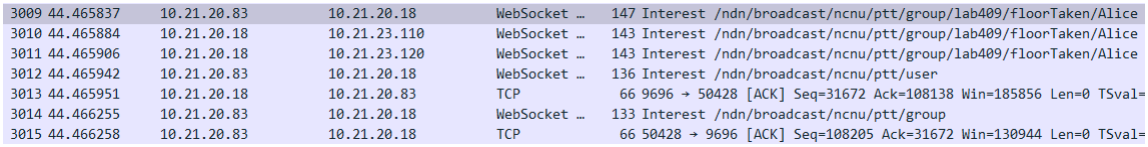
(監聽 Tunnel 的封包)



1	0.000000	10.21.20.83	10.21.20.18	TCP	78	50428 → 9696 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=40192
2	0.000019	10.21.20.18	10.21.20.83	TCP	74	9696 → 50428 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_P
3	0.001412	10.21.20.83	10.21.20.18	TCP	66	50428 → 9696 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=40192145 TSec
4	0.001585	10.21.20.83	10.21.20.18	HTTP	551	GET / HTTP/1.1
5	0.001590	10.21.20.18	10.21.20.83	TCP	66	9696 → 50428 [ACK] Seq=1 Ack=486 Win=30080 Len=0 TSval=2342979507 T
6	0.001661	10.21.20.18	10.21.20.83	HTTP	222	HTTP/1.1 101 Switching Protocols
7	0.003082	10.21.20.83	10.21.20.18	TCP	66	50428 → 9696 [ACK] Seq=486 Ack=157 Win=131584 Len=0 TSval=40192146

圖七十二、WebSocket 連線建立

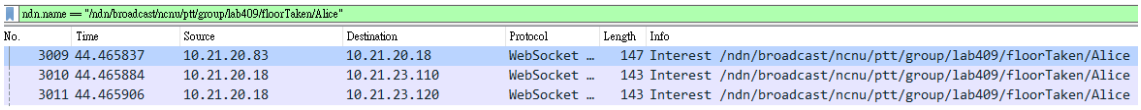
NDN 網頁即按即說應用會在 TCP Port 9696 上建立的 TCP 連線，接著申請將此 TCP 連線升級成 WebSocket 連線，並以此 WebSocket 連線作為 Tunnel，傳遞 NDN 封包，如圖七十二。



3009	44.465837	10.21.20.83	10.21.20.18	WebSocket …	147	Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice
3010	44.465884	10.21.20.18	10.21.23.110	WebSocket …	143	Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice
3011	44.465906	10.21.20.18	10.21.23.120	WebSocket …	143	Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice
3012	44.465942	10.21.20.83	10.21.20.18	WebSocket …	136	Interest /ndn/broadcast/ncnu/ptt/user
3013	44.465951	10.21.20.18	10.21.20.83	TCP	66	9696 → 50428 [ACK] Seq=31672 Ack=108138 Win=185856 Len=0 TSval=
3014	44.466255	10.21.20.83	10.21.20.18	WebSocket …	133	Interest /ndn/broadcast/ncnu/ptt/group
3015	44.466258	10.21.20.83	10.21.20.18	TCP	66	50428 → 9696 [ACK] Seq=108205 Ack=31672 Win=130944 Len=0 TSval=

圖七十三、Alice 按下發話按鈕

假設現有三個使用者 Alice、Bob、Carol 進行談話，Alice 按下發話按鈕，如圖七十三所示。Alice (10.21.20.83) 會發送 Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice (封包編號 5009、5010、5011)，通知群組內的 Bob (10.21.23.110) 與 Carol (10.21.23.120)，發話權已經被 Alice 取走。



ndn.name == "/ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice"

No.	Time	Source	Destination	Protocol	Length	Info
3009	44.465837	10.21.20.83	10.21.20.18	WebSocket …	147	Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice
3010	44.465884	10.21.20.18	10.21.23.110	WebSocket …	143	Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice
3011	44.465906	10.21.20.18	10.21.23.120	WebSocket …	143	Interest /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice

圖七十四、使用 Display Filter

其中，封包編號 5013~5015 是搜尋使用者的 Interest 或 TCP ACK，如果只

doi:10.6837/ncnu201900270

想觀察取得發話權的封包過程。這時，可以在 Display Filter 輸入

>> ndn.name == "/ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice"

將所有 Name 為 /ndn/broadcast/ncnu/ptt/group/lab409/floorTaken/Alice 的 NDN 封包過濾出來 (需新增 Wireshark 對 NDN 封包的解析)，如圖七十四所示。

3021	44.471077	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/0
3022	44.471081	10.21.20.18	10.21.23.110	WebSocket …	195	Interest /ndn/broadcast/ncnu/ptt/group
3023	44.471127	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/0
3024	44.471711	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/1
3025	44.472047	10.21.20.83	10.21.20.18	TCP	66	50428 → 9696 [ACK] Seq=108903 Ack=31929 Win=130944 Len=0 TSval=
3026	44.472050	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/1
3027	44.472212	10.21.20.83	10.21.20.18	WebSocket …	414	Data /ndn/broadcast/ncnu/ptt/user/Alice
3028	44.472921	10.21.20.83	10.21.20.18	TCP	66	50428 → 9696 [ACK] Seq=109251 Ack=31994 Win=130944 Len=0 TSval=
3029	44.473742	10.21.20.83	10.21.20.18	WebSocket …	416	Data /ndn/broadcast/ncnu/ptt/group/lab409
3030	44.473745	10.21.20.18	10.21.20.83	TCP	66	9696 → 50428 [ACK] Seq=31994 Ack=109601 Win=185856 Len=0 TSval=
3031	44.473752	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/2
3032	44.473777	10.21.20.18	10.21.23.110	TCP	66	9696 → 55730 [ACK] Seq=21587 Ack=98484 Win=228608 Len=0 TSval=
3033	44.473811	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/2
3034	44.474768	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/0
3035	44.474773	10.21.20.18	10.21.23.120	WebSocket …	195	Interest /ndn/broadcast/ncnu/ptt/group
3036	44.474847	10.21.20.83	10.21.20.18	TCP	66	50428 → 9696 [ACK] Seq=109601 Ack=32059 Win=130944 Len=0 TSval=
3037	44.477018	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/1
3038	44.477723	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/2
3039	44.477732	10.21.20.18	10.21.23.120	TCP	66	9696 → 42190 [ACK] Seq=16100 Ack=60839 Win=185856 Len=0 TSval=3
3040	44.492062	10.21.23.110	10.21.20.18	WebSocket …	412	Data /ndn/broadcast/ncnu/ptt/user/Bob
3041	44.492722	10.21.23.110	10.21.20.18	WebSocket …	416	Data /ndn/broadcast/ncnu/ptt/group/lab409
3042	44.492726	10.21.20.18	10.21.23.110	TCP	66	9696 → 55730 [ACK] Seq=21587 Ack=99180 Win=228608 Len=0 TSval=
3043	44.493621	10.21.23.120	10.21.20.18	WebSocket …	414	Data /ndn/broadcast/ncnu/ptt/user/Carol
3044	44.495880	10.21.23.120	10.21.20.18	WebSocket …	416	Data /ndn/broadcast/ncnu/ptt/group/lab409

圖七十五、要求語音訊息

接下來 Bob 與 Carol 會開始向 Alice 要求語音訊息，如圖七十五所示。這部分的封包資料會摻雜許多搜尋使用者與 TCP ACK 的封包，導致難以觀察語音訊息的封包。

ndn.namecomponent == "audio"

No.	Time	Source	Destination	Protocol	Length	Info
3021	44.471077	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/0
3023	44.471127	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/0
3024	44.471711	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/1
3026	44.472050	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/1
3031	44.473752	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/2
3033	44.473811	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/2
3034	44.474768	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/0
3037	44.477018	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/1
3038	44.477723	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/2
3122	45.145231	10.21.20.83	10.21.20.18	WebSocket …	813	Data /ncnu/ptt/group/lab409/user/Alice/audio/0
3128	45.145873	10.21.20.18	10.21.23.120	WebSocket …	872	Data /ncnu/ptt/group/lab409/user/Alice/audio/0
3130	45.146040	10.21.20.18	10.21.23.110	WebSocket …	872	Data /ncnu/ptt/group/lab409/user/Alice/audio/0
3140	45.164152	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/3
3142	45.164212	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/3
3145	45.169982	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/3
3201	45.546852	10.21.20.83	10.21.20.18	WebSocket …	813	Data /ncnu/ptt/group/lab409/user/Alice/audio/1
3203	45.546984	10.21.20.18	10.21.23.120	WebSocket …	809	Data /ncnu/ptt/group/lab409/user/Alice/audio/1
3205	45.547013	10.21.20.18	10.21.23.110	WebSocket …	809	Data /ncnu/ptt/group/lab409/user/Alice/audio/1
3211	45.549970	10.21.23.120	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/4
3212	45.550014	10.21.20.18	10.21.20.83	WebSocket …	131	Interest /ncnu/ptt/group/lab409/user/Alice/audio/4
3217	45.553293	10.21.23.110	10.21.20.18	WebSocket …	135	Interest /ncnu/ptt/group/lab409/user/Alice/audio/4

圖七十六、要求語音訊息 (過濾後)

這時可以在 Display Filter 輸入

>> ndn.namecomponent == "audio"

將所有 Name 中包含 audio 的 NDN 封包過濾出來，如圖七十六所示。

doi:10.6837/ncnu201900270

NDN 網頁即按即說應用會預先要求三份的語音訊息資料，在這裡，Bob (10.21.23.110) 的 Interest 先到達了 NDN Node (10.21.20.18)，並且已送往 Alice (10.21.20.83) (封包編號3021~3033)。當 Carol (10.21.23.120) 的 Interest 也到達 NDN Node 後 (封包編號 3034~3038)，會發現先前 Bob 的 Interest 已被記錄在 PIT 了。因此，Carol 的 Interface 會直接被記錄進 PIT，而不會送往 Alice。

在這次的語音訊息要求中，雖然 Bob 與 Carol 都有要求，但 Alice 只會收到一份 Interest。接著，NDN Node 在收到 Alice 回傳的 Data 後，會根據 PIT 的記錄，自動將 Data 複製成多份並傳遞給 Bob 與 Carol (封包編號 3122,3128,3130 以及 3201,3203,3205)，此為 NDN 的 Multicast 機制。

實驗結果顯示，將 NDN 應用在網頁即時通訊以及網頁即按即說上，均可達到所需的功能，如：註冊、搜尋、建立會話、文字訊息交換、語音訊息交換等。而在實作的過程中，NDN 也展現了網路節點如何快取資料內容，節省相同的資料要求。此外，當遇到相同要求的時候，網路節點能自動以 Multicast 的方式進行回傳。證明 NDN 是值得被推廣的網路架構。

doi:10.6837/ncnu201900270

第六章 結論及未來展望

整體網路流量有很大的部分是 Video Streaming 與 Webpage 的封包，若是能減少其中重複內容的封包，可望大幅降低網路雍塞的程度。本研究認為 NDN 在這方面上，有著非常適合的機制存在，如: 網路節點的快取機制。但，目前 NDN 在瀏覽器的應用上仍相當缺乏，使用者若是只能瀏覽網頁卻沒有其他功能可以使用，可能會認為，不如就使用原本 IP 網路中的瀏覽器 (Chrome、Safari、Firefox) 就好。本研究透過在 NDN Browser 的基礎上，實作出運行在 NFD 上的 Web Consumer，並且提供網頁即時通訊的應用，讓接觸到 NDN 的人，也可以使用到實用的工具，達到吸引到更多使用者以及推廣 NDN 的效果。

本研究中的 Web Consumer 以單一執行緒撰寫，尚無法以多工的方式處理 HTML 標籤，且仍有許多功能需要補強。此外，Web Consumer 與 Web Producer 之間的溝通方式也需要制定一個完善的標準，讓這兩者可以持續成長到如同 Chrome 瀏覽器與 Apache 伺服器那般完整的功能。

現階段的語音通訊品質仍未臻理想，其中，利用 RecordRTC 所錄製的語音訊息，長度為 350 ms (低於 250 ms 會造成聲音資料無法錄製)。因為錄製長度的關係，傳遞語音訊息的時間間隔亦為 350 ms (錄製完才送出)，無法達到 IP 網路的標準 (通常為 20~50 ms 就送出一個音訊封包)。未來可以開發一些降低延遲的功能以及更好的語音傳輸方式，使其可以有更好的通話品質。

另外，在 NDN 網頁即按即說應用中，目前若是同群組內多人同時按下按鈕時，該如何解決衝突，也是未來必須解決的問題。

doi:10.6837/ncnu201900270

參考資料

[1] CloudFlare, "What Is A Reverse Proxy?" [Online] Available: https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/

[2] CoudFlare, "What Is a CDN?" [Online] Available: https://www.cloudflare.com/learning/cdn/what-is-a-cdn/

[3] Eric Bina and Marc Andreessen, "Mosaic Browser", *History of Computers*. [Online] Available: https://history-computer.com/Internet/Conquering/Mosaic.html

[4] X. Qiao, G. Nan, Y. Peng, L. Guo, J. Chen, Y. Sun and J. Chen, "NDNBrowser: An extended web browser for named data networking", *J. Netw. Comput. Appl.*, pp. 1-14, 2014.

[5] Junxiao Shi, "Browsing in NDN" [Online] Available: https://www.lists.cs.ucla.edu/pipermail/ndn-interest/2019-July/002520.html

[6] Wentao Shang et al., "Firefox Add-on: Firefox Add-on for the NDN Protocol" [Online] Available: https://github.com/named-data/ndn-js

[7] Wikipedia, "CCNx". [Online] Available: https://en.wikipedia.org/wiki/Content_centric_networking

[8] PARC, "About Palo Alto Research Center". [Online] Available: https://www.parc.com/about-parc/

[9] Alexander Afanasyev et al., "NFD Overview". [Online] Available: http://named-data.net/doc/NFD/current/overview.html

[10] Wikipedia, "Web Cache Communication Protocol". [Online] Available: https://en.wikipedia.org/wiki/Web_Cache_Communication_Protocol

[11] Wikipedia, "DVMRP (Distance Vector Multicast Routing Protocol)". [Online] Available: https://en.wikipedia.org/wiki/Distance_Vector_Multicast_Routing_Protocol

[12] J. Moy, "MOSPF: Analysis and Experience", *IETF RFC 1585*, March 1994. http://www.ietf.org/rfc/rfc1585.txt

[13] Cisco, "PIM Snooping". [Online] Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/snooppim.html#wp1006510

[14] "WebRTC - Web Real-Time Communication ". [Online] Available: https://webrtc.org/

doi:10.6837/ncnu201900270

[15] Michael Herrmann, "PyQt5 Documentation" [Online] Available: https://wiki.python.org/moin/PyQt

[16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", *IETF RFC 3261*, June 2002. http://www.rfc-editor.org/rfc/rfc3261.txt.

[17] Z. Zhenkai, W. Sen Y. Xu, V. Jacobson and Z. Lixia, "ACT: Audio Conference Tool Over Named Data Networking", Proceedings of the ACM SIGCOMM workshop on Information-centric networking (ICN), pp.68-73, Toronto, Canada, 2011.

[18] Peter Gusev and Jeff Burke, "NDN-RTC: Real-time videoconferencing over named data networking", Proceedings of the 2nd ACM Conference on Information-Centric Networking (ICN), pp. 117–126, San Francisco, USA, 2015.

[19] L. Wang, I. Moiseenko and L. Zhang, "NDNLive and NDNTube: Live and prerecorded video streaming over NDN," Technical Report NDN-0031, 2015.

[20] W. Shang, J. Thompson, M. Cherkaoui, J. Burkey and L. Zhang, "NDN.JS: A JavaScript client library for named data networking," 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) , pp. 399-404, Turin, Italy, 2013.

[21] Muaz Khan, "Audio+Video+Screen Recording using RecordRTC" [Online] Available: https://www.webrtc-experiment.com/RecordRTC/

[22] Davide Pesavento, "ndn-tools" [Online] Available: https://github.com/named-data/ndn-tools

[23] "PyNDN documentation!" [Online] Available: http://named-data.net/doc/0.3.2/PyNDN2/

[24] Qt Company Ltd., "Qt WebEngine Overview". [Online] Available: https://doc.qt.io/qt-5/qtwebengine-overview.html

[25] Y. Wu, Q. Wu, "Push to Talk (PTT) using Multicast UDP with IPv6", Proceedings of the Taiwan Academic Network Conference (TANet2019), Kaohsiung, Taiwan, September 25-27, 2019.

[26] Chien-Maw Jen, "Method for implementing push-to-talk over SIP and multicast RTP related system" [Online] Available: https://patents.google.com/patent/US7620413B2/en

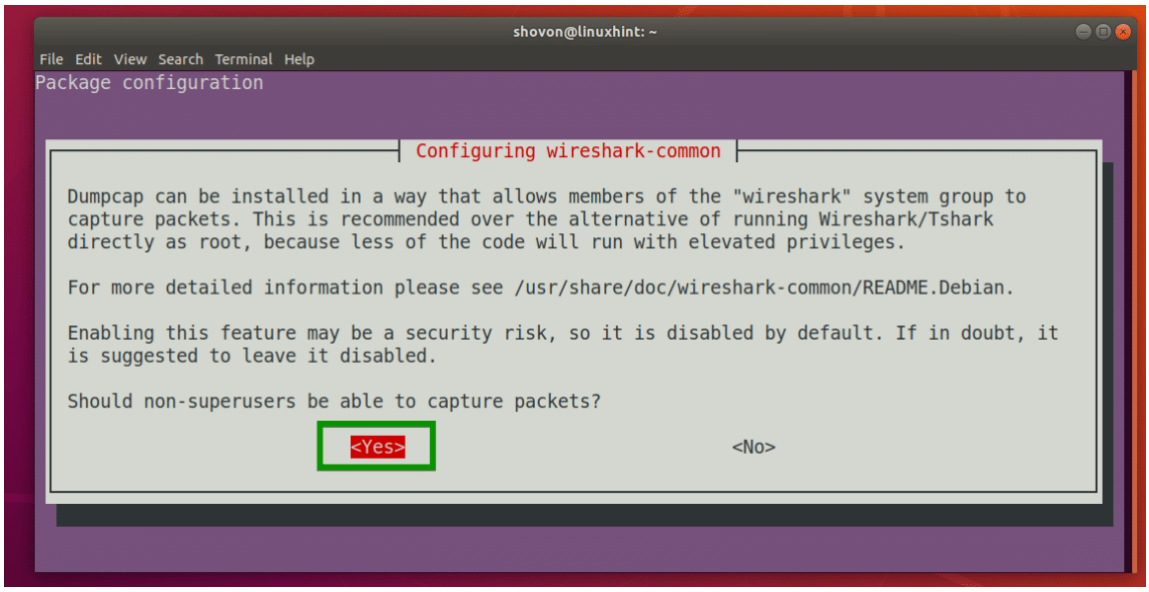
doi:10.6837/ncnu201900270

附錄

附錄一 Wireshark 安裝

Ubuntu 18.04 可以直接使用 apt 進行安裝。

>> sudo apt install wireshark



附圖一、允許 non-superuser 使用

安裝過程會詢問是否允許 non-superuser 使用，請選擇 Yes，如圖七十七所示。

安裝完成後，將使用者添增 wireshark 的 group。

>> sudo usermod -a -G wireshark user_id

重啟電腦，即可使用 Wireshark 抓取封包。

doi:10.6837/ncnu201900270

附錄二 NDN Multicast Suppression Duration

NFD 上有個 Suppression Duration 政策，假設有兩名 Web Consumer 發出了網頁要求，NFD 會將收到的 Interest 做 Aggregate，也就是利用 Pending Interest Table (PIT) 記錄 Interest 進入的介面，只會送出一份 Interest 到 Web Producer。這是理想的運作，可是受到 Suppression Duration [21] 的影響，若兩份 Interest 到達 NFD 時間間隔大於 10 ms (此時間間隔會因為 Retransmit 而倍數增加，最大至 250 ms)，NFD 就不會 Aggregate，而是將兩份 Interest 都送到 Producer 上。第一份 Data 回傳後，NFD 仍會 Multicast 到兩名 Consumer 手上，並且忽略掉第二份 Retransmit 的 Data。

doi:10.6837/ncnu201900270