目錄

摘要

命名資料網路（Named Data Networking, NDN）[2]是以解決現今人們對於網路的需求為目標而設計的未來網路架構。以資料中心網路（Content-Centric Networking, CCN）[1]為基礎，延續其以資料作為中心的概念，它能解決 IP 協定在人們對於網路的需求下遭遇的問題與困難，例如：使用者在網路中的匿名性、資料傳輸的安全性、資料取得的路徑、傳輸過程中的封包暫存、IP 位址空間耗盡等問題。儘管帶著許多優勢，NDN 仍是尚未完善的網路架構，許多方面，例如：公鑰取得方式、封包命名標準等皆尚未有定論，加深了實現其的難度。

為了解決上述 IP 遭遇的問題，建設更符合現今需求的網路環境，在本次專題製作，我們會以 NDN 的理論為主，以 CCN 的概念為輔，配合密鑰生成中心（Key Generation Center, KGC）生成可信任公鑰與私鑰，實作出 NDN 網路架構，並在其以資料為中心的基礎上，期望可以加入新的特性與細節，使之更為完善。

關鍵詞：命名資料網路、匿名性、數位簽章、密鑰生成中心

Abstract

Named Data Networking (NDN) is a novel Internet architecture that meets people's requirements for Internet use today. NDN based on Content-Centric Networking (CCN) which focus on data-centric transmission instead of host-centric communication, can deal with problems that people encounter while using today's Internet, including anonymity of Internet users, content security while transmitting packets, the path of transmitting packet, content caching in routers, and the state of running out of IP address. Even though NDN possesses advantages over IP in many aspects, it is still an incomplete Internet architecture, and many details of NDN have not been conclusive yet. For instance, the process of obtaining public keys and the standard of naming contents.

To resolve the problems that people encounter with the Internet and construct an Internet environment that meets Internet use today, we decided to implement NDN architecture and generate trusted keys by Key Generation Center (KGC). Besides, we expect to add some new details and features to NDN and make it more complete.

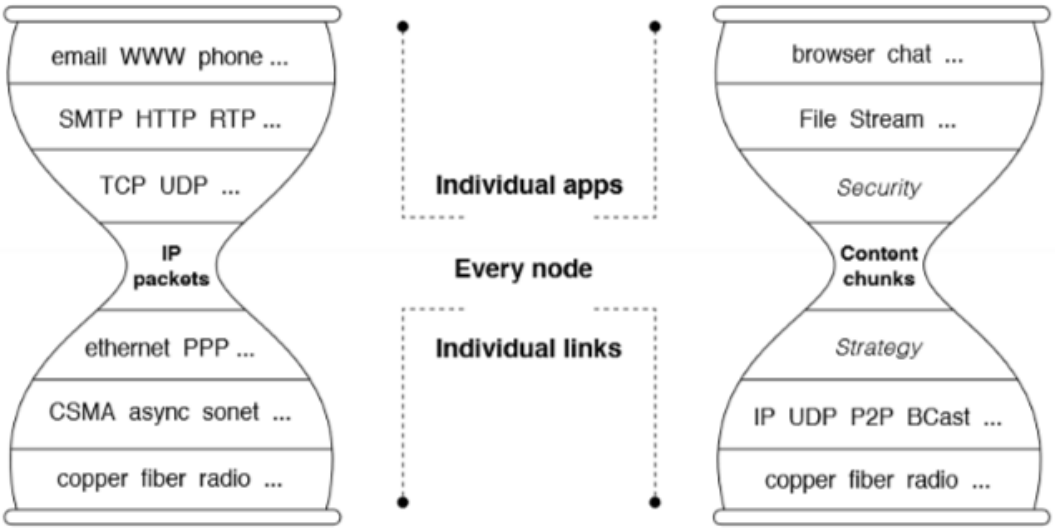
Keyword: Named Data Networking, Anonymity, Digital Signature, Key Generation Center

章節 1

簡介

現在人們使用網路的網路層協定主要是奠基於 IP－以主機為中心的網路架構，所有資料的取得途徑都是連接至擁有該資料的主機並索取資料。在網路的發展下，伴隨著物聯網、社群網路等的出現，我們對於網路的使用及需求已漸漸傾向於去中心化網路。在學習 IP 網路的運作後，我們了解到：現今網路架構仍存在著許多問題。舉例來說：在 IP 網路中，我們需要知道資料擁有者的 IP 位址，才能進行連線並取得資料，但也因此，資料擁有者的 IP 位址可以被任何人知道，使其身分暴露在網路中、造成匿名安全性問題；再舉一個例子：我們經常需要藉由社群網路發布自己的資料（或者檔案）或接收其他人的資料，此時我們需要先連接至離自己地理位置較遠的社群網路服務伺服器，再從伺服器連接至欲溝通的人，儘管我們處在同一區域網路中，或甚至就在彼此身邊。

為了解決上述問題，Van Jacobson 等人提出了命名資料網路（Named Data Networking, NDN）[2]這個未來的網路架構，NDN 提倡以資料本身為中心：將封包命名，並以包含其名稱的請求來找尋網路中對應的資料，不同於 TCP/IP 網路經由 IP 位址標示封包傳輸的目的地，換句話說，使用者要求檔案的方式是告訴網路要求檔案的名稱，而不是告訴其要傳送的目的位置。在 OSI 模型中，儘管 NDN 在網路架構第四層中取代了 IP 了位置（圖一），卻保留了 TCP/IP 網路細腰架構的優勢－讓檔案經由頂層到底層時必定會經過 IP 協定，反之亦然，使得網路架構各層的設計更加容易，也造就現今網路的順利發展。除了 OSI 模型第四層之外，NDN 其上下兩層的協定分層也與 TCP/IP 網路不同：分別是 Security 和 Strategy。簡要來說，Security 層使得資訊安全安保護可以在資料本身上實行，而不像 IP 需要在兩個主機間的整個傳輸過程中進行維護，Strategy 層則讓網路層更利於與第二層同時建立多種類型的連線。



圖一（IP 與 NDN 的協定分層架構[1]）

在 NDN 中，封包被區分為興趣封包與資料封包，興趣封包可供檔案要求者標示要求的檔案名稱，資料封包則是包含著檔案內容並讓檔案生產者傳至網路中。NDN 路由器可經特定演算法決定是否將資料封包暫存至暫存區中，而興趣封包則可經由路由器轉發至暫存著對應資料封包的路由器，再將其對應資料封包回傳至檔案要求者。我們將在章節二中更詳細的說明 NDN 運作方式。

1.1 研究動機

在認識到 NDN 的架構後，我們了解到其以資料為中心的特性更符合今日使用者的需求，在此網路架構下，凡舉上述資料擁有者匿名安全性問題與封包傳輸路徑的例子，抑或是現今 IP 網路遭遇的其他問題，例如：IP 位址空間不足、在應用層才能外加安全機制保護資料的傳輸等，都能被解決或改善。因此在本次專題研究計畫中，我們將實作並模擬 NDN 網路的運作，藉由其特性，解決現今網路遭遇的問題與困難，並嘗試讓 NDN 的架構更為完善，使其更接近人們可使用並依賴的網路架構。

1.2 研究目標

在本次專題研究計畫，我們希望在模擬 NDN 的同時，展示它與 IP 之間的明顯差異，並證明它是更符合於現今需求的網路架構。在下方我們列出本次專題實作的目標與期望。

(1) 解決 IP 位址暴露問題：由於在 NDN 中沒有 IP 位址的概念，使用者間不會知道互相的 IP 位址，在封包上也沒有附與 IP 相

關的任何訊息，我們在本次專題製作中用 TCP socket 去模擬 NDN，在實作上使用者只會知道其相鄰的（與之連接）路由器 IP 位址，不同於 IP 中，使用者會知道其所有溝通對象的 IP 位址。

(2) 資料傳輸安全性：NDN 對封包進行的安全保護（包含機密性、完整性與不可否認性）是基於附在封包上的數位簽章，不同於 IP，需要在兩個端點的應用層上進行對封包的安全保護。

(3) 傳輸過程中的封包暫存：路由器處理到達的封包並決定其轉發方向時，透過演算法決定是否將封包暫存至暫存區中，不同於 IP 路由器，在處理完封包的轉發後就刪除它。

(4) 取得較合適的資料路徑：因封包在路由器的暫存，路由器可轉發對於檔案的請求，並在尋找到暫存目標檔案的路由器後回傳目標檔案，將其以檔案請求的原轉發路徑傳輸至檔案要求者，相比於 IP 網路，要求者需先連接至檔案生產者（若不知道檔案生產者的 IP 位址，還需先連接至提供瀏覽器服務的伺服器），方可要求檔案。

(5) 解決 IP 位址空間不足：檔案要求者會在封包的名稱欄位上填入要求檔案的名稱，並送至網路中尋找擁有對應名稱的檔案，而其名稱欄位在本次專題實作中我們配置 32 位元組的空間，相比於 IP 網路的 IP 位址不管 IPv4 抑或是 IPv6 都有位址空間耗盡問題，NDN 的封包名稱不會有空間的限制。

章節 2

研究理論

第 2 章將說明本次專題製作奠基於的理論架構，其中包含 NDN 的架構（章節 2.1）、數位簽章（章節 2.2）與 KGC 和可信任的公鑰（章節 2.3）。

2.1 NDN 架構

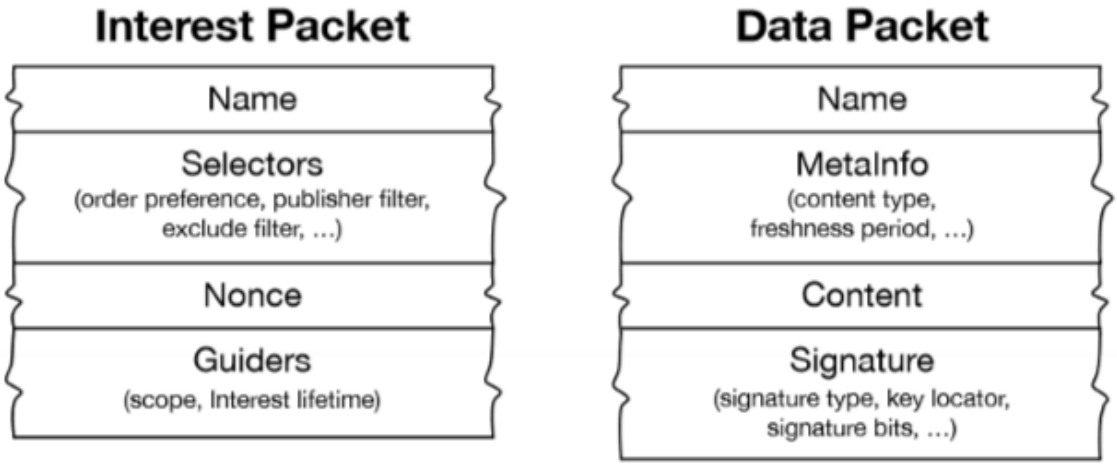
命名資料網路[1,2]是一個正在發展中的未來網路架構之一，相較於現今的 IP 網路以使用者做為中心，專注在”Where”－資料在哪裡，命名資料網路是一個以資料作為中心的網路結構，專注在”What”－資料本身。命名資料網路取得資料的來源可以是任何地方，不像傳統 IP 網路取得資料只能侷限在終端設備中。

在傳統的 IP 網路中，封包的傳輸方向是由其中的 IP 位置來決定，但在命名資料網路中則是由封包中的 Name 來決定，因此這個新的網路架構的封包設計會和傳統的不一樣。命名資料網路中有兩種封包：興趣封包（Interest packet）及資料封包（Data packet），兩者有各自不同的封包欄位（圖二）。

- **興趣封包**：包含了 Name（想要搜尋的資料名稱）、Selector（設定資料的範圍及其定義）、Nonce（防止封包迴圈的發生）等欄位。使用者會將想搜尋的資料名稱放入封包中傳入命名資料網路，而路由器則會經由 Name 將興趣封包轉發到檔案提供者或暫存著資料封包的路由器。
- **資料封包**：包含了 Name（資料名稱）、MetaInfo（主要設定資料的有效時間）、Content（資料的內容）、Signature（數位簽章等資訊）。當興趣封包傳輸到檔案提供者或暫存著對應資料封包的路由器時，會將對應的資料封包以興趣封包傳來的原路徑以反方向傳回去到使用者，並會在封包上加入數位簽章以達成資訊安全的不可否認性及完整性。

除此之外，命名資料網路的封包有兩種命名原則，第一種是直接使用與內容語意上相關或使用者可直接理解的字詞，例如：一部電影名稱或某人的名字。第二種為階層式的形式命名，可以經由不同長度

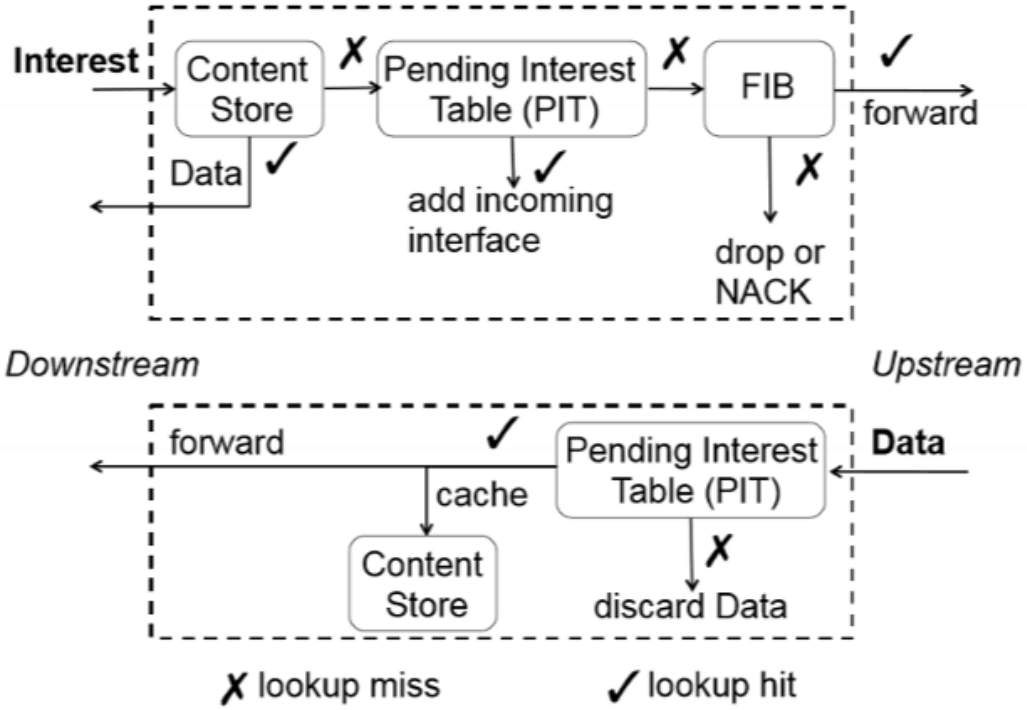
的字串組成，例如：中山大學資工系學生程式作業的檔案可以命名為”/NSYSU/CSE/student/homework.cpp”，其命名的原則跟 URL 十分相似。



圖二（封包格式[2]）

NDN 路由器需要一個轉發策略（Forwarding Strategy Module）以及三個資料結構（Pending Interest Table, Forwarding Information Base, Content Store）來達成完整的資料傳輸。

- **Content Store (CS)**：用於暫存資料封包，提供其他使用者可以直接取得資料。是否暫存曾經轉發過的資料封包，取決於路由器本身的機制，並且盡可能的留下重複使用率較高的資料，例如：當日新聞或熱門影片。
- **Pending Interest Table (PIT)**：紀錄尚未被滿足的興趣封包及其來源。當興趣封包被路由器轉發時，會紀錄其資訊在 PIT 條目中，而當資料封包回傳至路由器時，會將資料封包回傳至興趣封包的來源，並將其資訊由 PIT 中移除。
- **Forwarding Information Base (FIB)**：用於查詢並轉發資料的潛在來源位置。命名資料網路本身可以避免迴圈，所以允許同時發送多個請求。
- **Forwarding Strategy Module (FSM)**：有一系列的封包轉發政策，在特殊情況下甚至可以直接丟棄興趣封包，例如：興趣封包被當作是 DoS 攻擊的一部分時。



圖三（路由器處理封包流程[2]）

如上方的圖三所示，當暫存器收到興趣封包時：第一步檢查 CS 中是否暫存著對應的資料封包，如果有就回傳資料封包；如果沒有，檢查此興趣封包的資訊是否列在 PIT 紀錄中，且記錄有相同 Name 的封包，如果有就只需佇列至 PIT 列表中，不用再轉發一次其封包；如果沒有相同的紀錄，則必須除了新增其資訊至 PIT 列表之外，查詢 FIB 並轉發興趣封包到資料潛在來源；若 FIB 還是找不到該封包的資料來源時，興趣封包會被丟棄。

當找到對應的興趣封包時，資料封包會被資料提供者回傳到路由器中，並會查詢 PIT 列表是否有使用者要求其資料，如果有就將資料封包繼續回傳至前一個使用者或路由器，將資料刪除自 PIT 列表，並將封包暫存至 CS 中；如果 PIT 中找不到相應的紀錄，代表可能等待的時間過長，已放棄此封包，這時路由器會丟棄這個封包。

2.2 數位簽章

訊息發布的目的是為了讓人們知道訊息，因此沒必要對訊息做加密，但是我們必須排除訊息被他人偽造，同時也要避免接收到假消息，因此才會有數位簽章[3,4,5]存在的必要。

數位簽章的目的是為了要確保我們收到資料封包的來源是可信的，不會是來自不明人士發送的可疑資料。在數位簽章中，訊息提供

者會傳送訊息與簽章兩個文件，而訊息接收者會在收到這兩個文件後，會去證明這個簽章是否屬於傳送者的，如果證實如此的話，就可以保證訊息的來源。

在數位簽章的過程中，訊息提供者會使用私鑰來對雜湊過後的訊息作簽名，然後公布對應的公鑰，而訊息接收者會用公鑰來對簽名做驗證，並將解密的結果與訊息做比較。如果比較結果正確的話，訊息就會被保留，否則就會直接拋棄這個訊息。

可靠的加密演算法可以增加數位簽章的安全性，在良好的數位簽章下，我們可以保證這個資料封包沒有被篡改(訊息完整性)、可以確認這個資料封包的來源(訊息確認性)、並且也保證訊息提供者曾經上傳過這個資料封包(不可否認性)，藉此來增加使用者的安全性。

2.3 KGC 與可信任公鑰

儘管有了簽章來保護資料的安全性，若無法取得正確、可信任的公鑰來認證簽章，也是徒勞無功，因此需要確保自己可以取得合法來源，換言之來源正確的公鑰。而 Key Generation Center（KGC）[6,7] 就是一個取得可信任公鑰的管道，KGC 負責生成所有用戶的私鑰與公鑰並將其儲存在自己的鑰匙清單中，且會利用安全通道將私鑰傳至用戶，用戶也可以向 KGC 要求其他用戶的公鑰，以此認證簽章。在 3.3 章節中我們會更詳細的解釋使用者取得可信任公鑰的流程。

章節 3

研究方法

我們將在第 3 章說明專題個部份的實作方式。在章節 3.1 中說明路由器如何處理各種封包（包含興趣封包、資料封包、測試封包等）、在章節 3.2 說明數位簽章的雜湊與簽名實作方式、在章節 3.3 說明獲得可信任公鑰並認證簽章的流程。

3.1 路由器封包處理

我們在封包內有一個 Type 欄位，我們利用這個欄位來判斷這個封包的用途，首先分為興趣封包(Type = 1)、資料封包(Type = 2)，由測試封包(Type = 3)來探測目前的連線裝況，而當路由器與其他路由器連線時，會先發送請求連線的封包(Type = 4)，如果連線成功的話，會回傳成功連線的封包(Type = 5)，連線失敗的話會回傳失敗封包(Type = -1)。

我們用 Nonce 欄位來判斷路由器是否已經接收過這個封包了，當收到興趣封包時，我們會先比對 PIT 內存放的興趣封包，如果裡面有名稱相同、 Nonce 的值也相同的話，就表示已經接收過相同來源的興趣封包了，路由器就會直接拋棄這個興趣封包。如果沒有相同的興趣封包的話，路由器會將這個封包與他的來源存放在 PIT 內。

當收到資料封包時，會先將資料封包存入 CS 內，然後查看 PIT 內是否有與這個資料封包相關的興趣封包，如果有的話會順著這個封包的來源回傳回去。

我們在路由器上額外開一個執行緒來計算興趣封包的存活時間。每隔固定的時間就會檢查 PIT 內是否有存放過久的興趣封包，並且把所有過期的興趣封包拋棄。

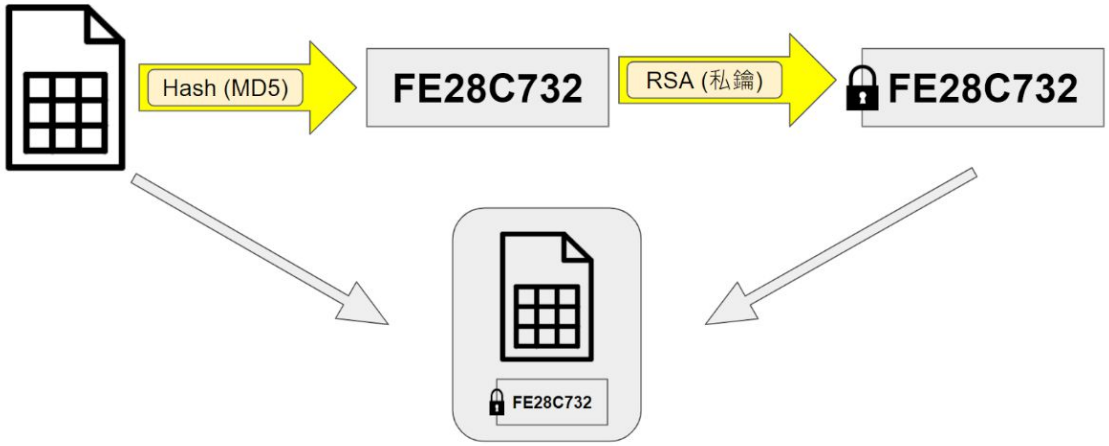
我們在存放資料封包時會使用 LRU 快取演算法(Least Recently Used) ，這種演算法會判斷最近使用的資料是熱門資料，有很大的機率會再一次被使用。每當路由器收到資料封包時會將資料封包放在 CS 內的第一個欄位，並且當 CS 的內存不夠用的時候，會將存放在 CS 內最後一個資料封包拋棄，如果 CS 內有資料被再次使用的話，

路由器會將這個資料在 CS 內的位置變更到 CS 內存的第一個欄位，藉此來降低此資料封包被拋棄的機率。

3.2 數位簽章之雜湊與簽名

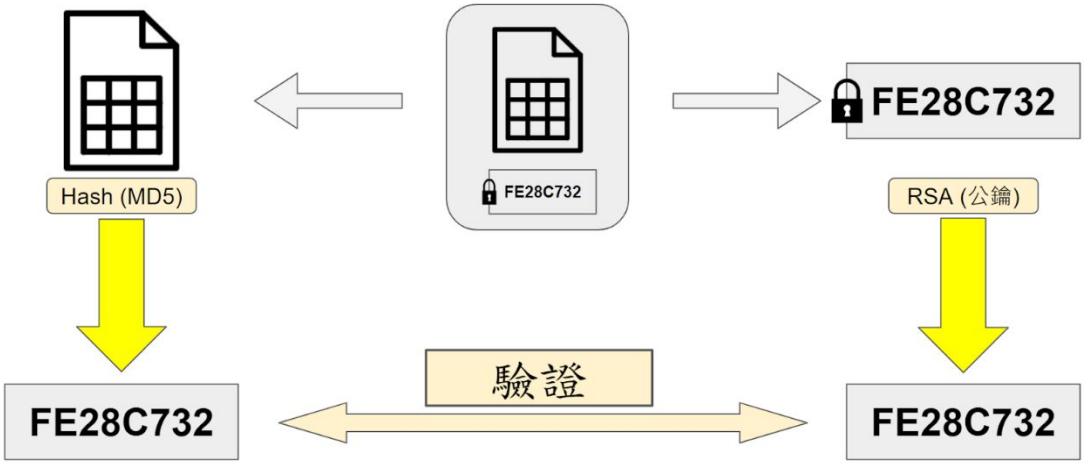
數位簽章的實作分為兩個部分：雜湊和加密，而在這次專題製作我們分別使用了 MD5 和 RSA 實作數位簽章，在 MD5 的部份使用網路上其他撰寫的程式碼[8]來執行，而 RSA 的部份我們實作了大數運算來實行 RSA 的計算，並利用快速冪次演算法和蒙哥馬利約分來進行加速，也因為沒有使用到更多的演算法，我們把金鑰的長度設為 248 bits。

我們實作數位簽章的流程如下：每當檔案提供者要送出新的資料封包前，首先會先用 MD5[5]將資料做雜湊，MD5 會將資料轉換成長度固定為 16 Bytes 的雜湊值，之後利用檔案提供者自己的私鑰用 RSA 加密演算法[4]把雜湊值做加密，加密後的值將會作為資料的簽名，並且簽名會與資料一起被放進資料封包裡面傳送出去（圖四）。



圖四（簽名流程）

當使用者接收到資料封包後，會先將資料封包拆分成資料與簽名兩部分，先用公鑰將簽名做 RSA 解密，然後將資料用 MD5 做雜湊，雜湊後的結果與 RSA 解密後的結果做驗證（圖五），如果結果相同，就可以驗證這個資料來自可靠的來源，如果結果不同，則檔案接收者會將此資料封包丟棄。

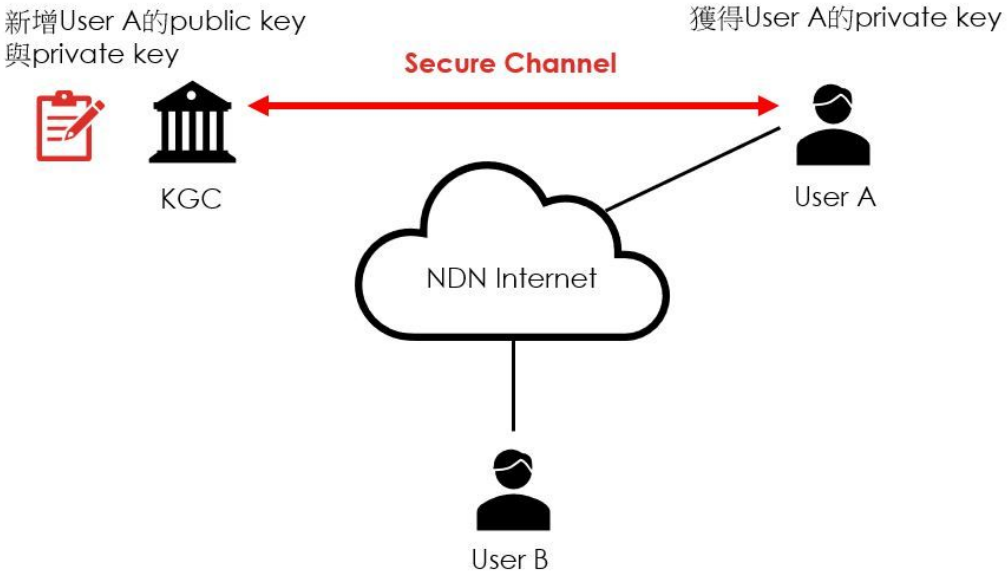


圖五（認證簽章）

3.3 公鑰取得與簽章認證

取得可信任公鑰並認證其合法來源是時做 NDN 中不可或缺的一環，少了它，儘管有簽章附在資料封包上做安全保護，檔案要求者還是有可能取得惡意使用者的公鑰，並認證惡意取相同名稱的封包而不自知。以下是配合 KGC 取得可信任公鑰並認證合法來源的流程。

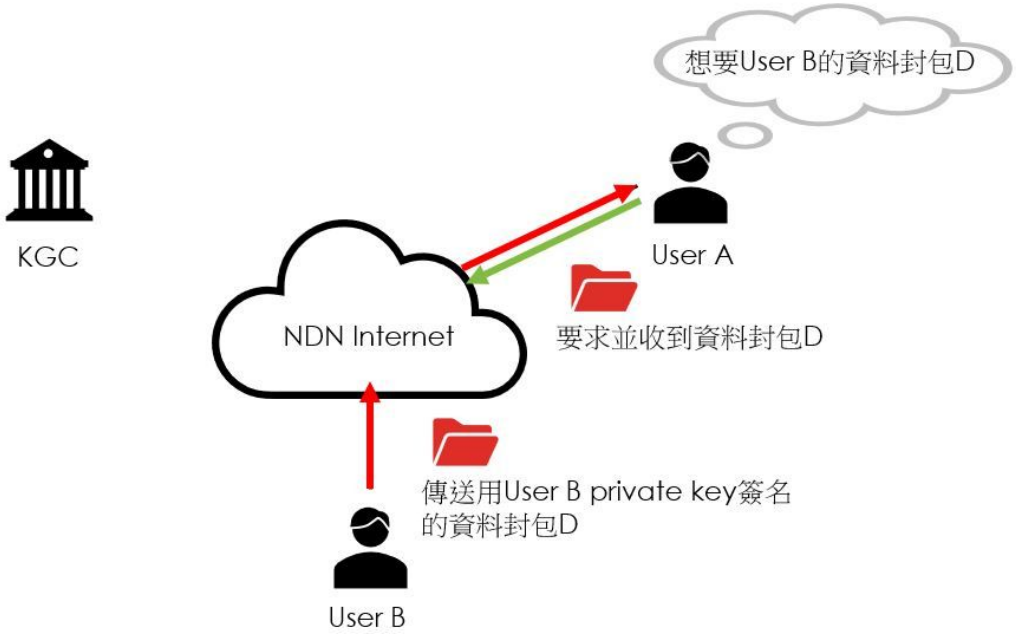
(1) 每一個使用者在加入網路前都會與 KGC 建立安全通道以交換訊息：KGC 生成 User A 的公鑰與私鑰且存至自己的列表中，並將其私鑰傳給 User A，安全通道可以像是使用者本人至臨櫃註冊使用此 NDN 網路，並取得自己的私鑰（圖六）。



圖六（建立安全通道）

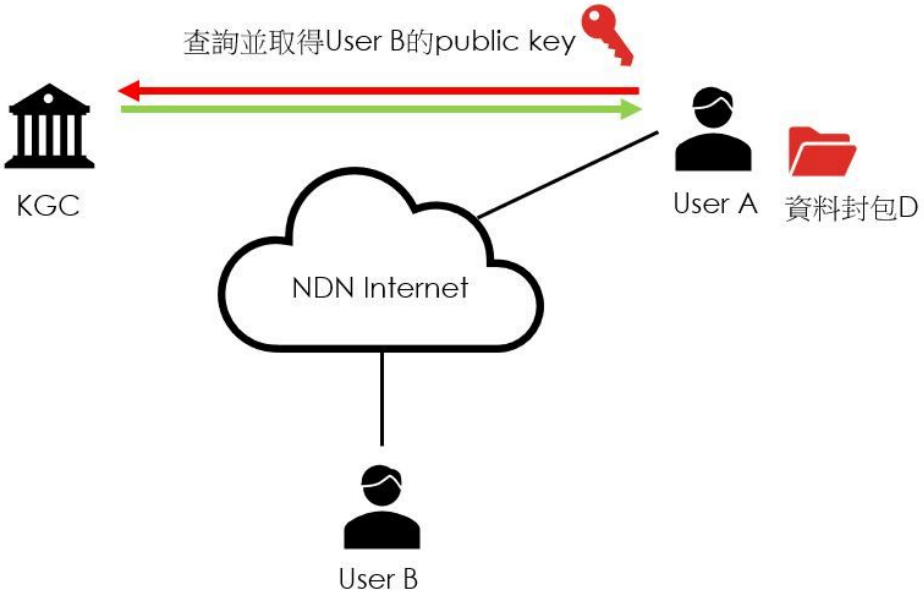
(2) 假設 User A 想要 User B 的資料 D（假設資料 D 僅有一個資料封包內容的大小），並已向 NDN 網路要求且取得資料 D，如圖

七，但 A 不能確定自己收到的資料是否為源自於 B，因為有可能另一位檔案生產者在刻意或無意間上傳了同樣名稱但內容不同的資料 D。



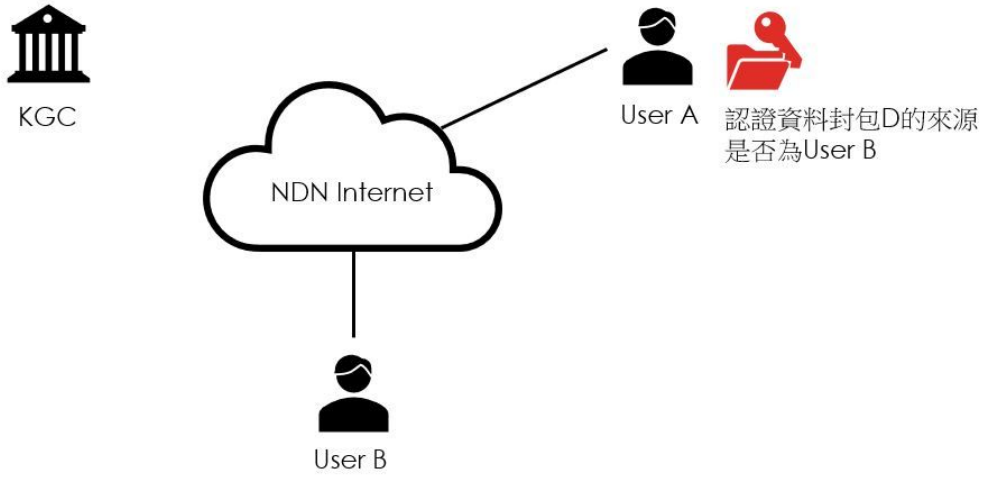
圖七（送出要求並收到封包）

(3) 在 NDN 的資料封包中都附著一個數位簽章，User A 認證此資料 D 是源自於 User B 的方式是：用 B 的公鑰認證附在資料 D 上的簽章，而取得可信任公鑰的方式是：直接跟 KGC 索取公鑰，在實作上，我們將使用者直接連接至 KGC 的 IP 位址並要求 User B 的公鑰，如圖八。



圖八（向 KGC 取得公鑰）

(4) 最後將取得的公鑰直接對收到的資料 D 做認證，確認資料 D 的來源是否為合法來源 User B，如圖九。



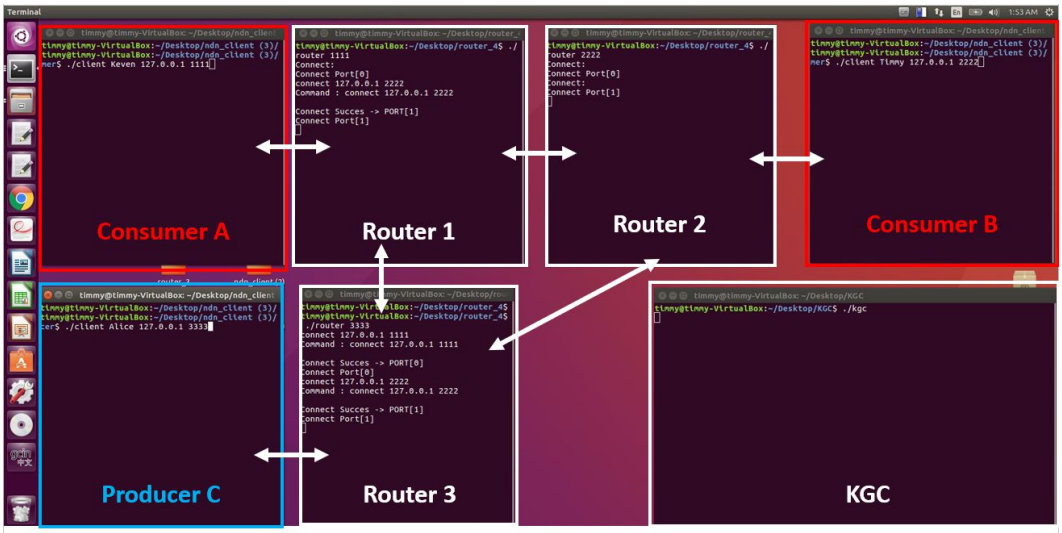
圖九（認證簽章）

章節 4

實作流程

在第 4 章中，我們會說明本次專題的實作步驟，4.1 解釋使用者如何加入至網路，4.2 至 4.4 說明檔案生產者傳檔案至網路、檔案要求者要求並取得檔案的步驟，4.5 則說明如何認證封包的合法來源。

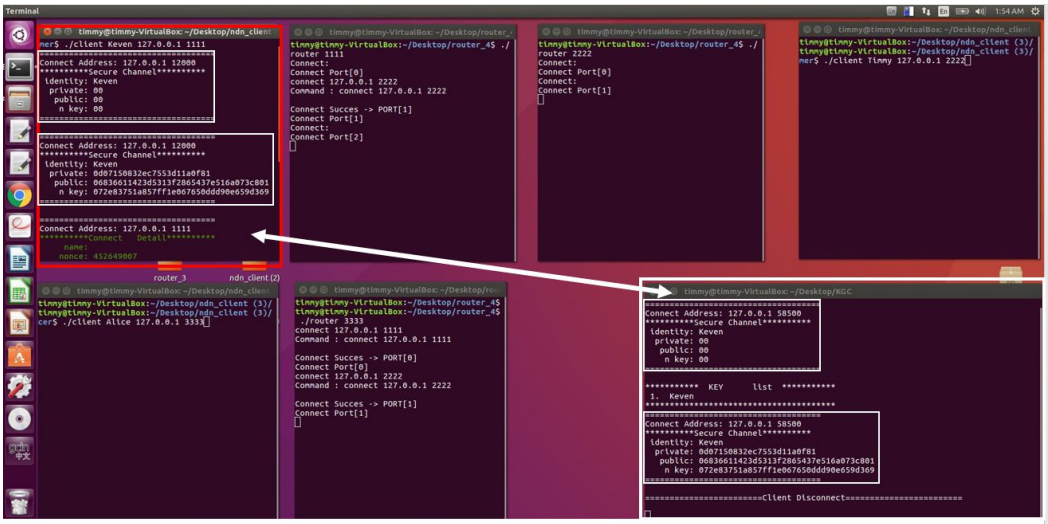
本次專題實作的網路我們以三個路由器、三個使用者以及 KGC 為範例進行展示，其連接方式如圖十，檔案要求者為紅色的框，檔案生產者為藍色的框，KGC 以及路由器皆為白色的框。



圖十（網路中路由器與使用者的連接方式）

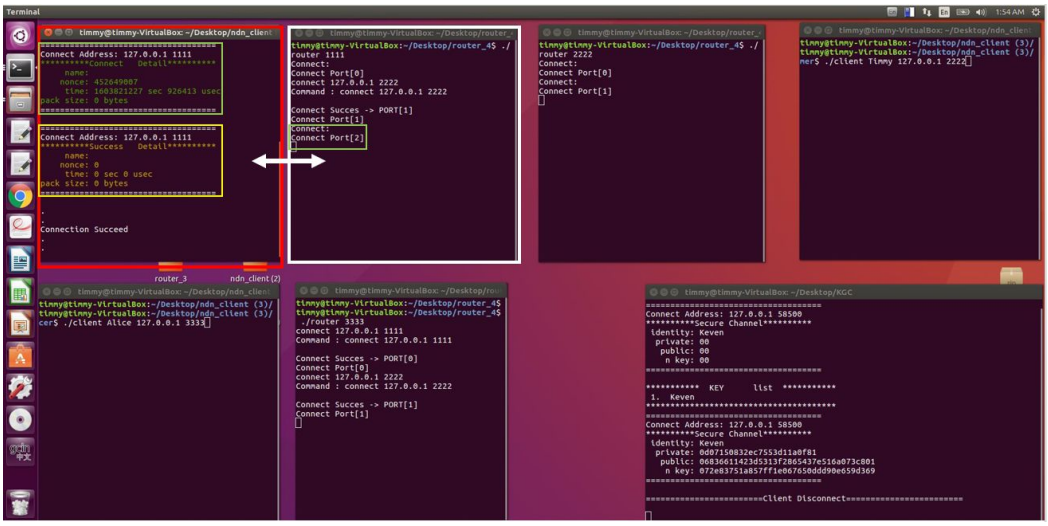
4.1 使用者加入網路

使用者與 KGC 建立安全通道交換資訊，KGC 紀錄使用者的私鑰與公鑰至自己的清單，使用者則從 KGC 取得自己的私鑰與公鑰。如圖十一中的 Consumer A 直接連接 KGC，與之溝通並取得自己的私鑰與公鑰（圖中的白色細框為包含著私鑰與公鑰的封包），KGC 則儲存 User A 的公鑰與私鑰在自己的清單中。



圖十一（與 KGC 交換資訊並取得私鑰）

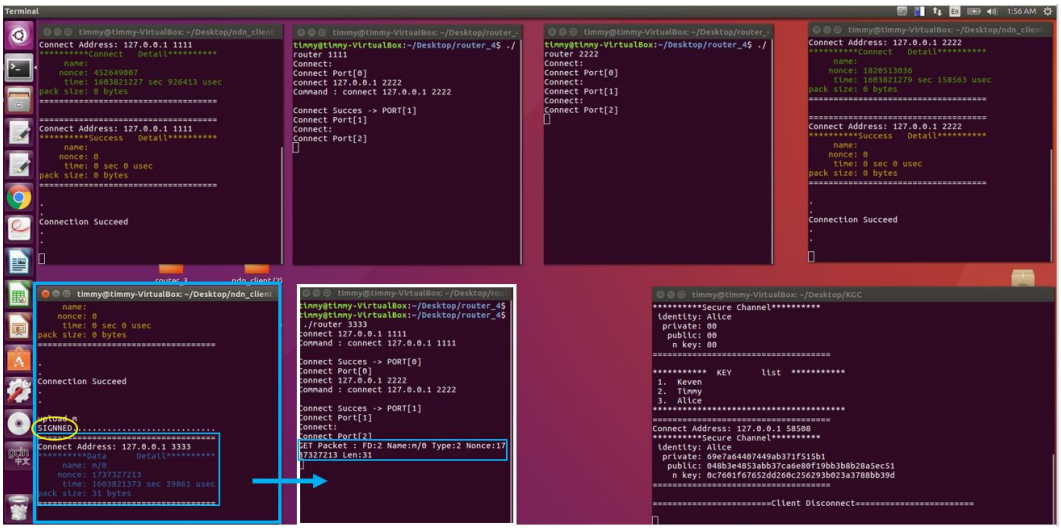
與 KGC 溝通後，使用者與最相鄰的路由器連接。如圖十二，Consumer A 與相鄰的 Router 1 連接（圖中綠色細框為要求連線的封包，黃色細框則為連線成功的封包）。



圖十二（與相鄰之路由器連接）

4.2 檔案上傳

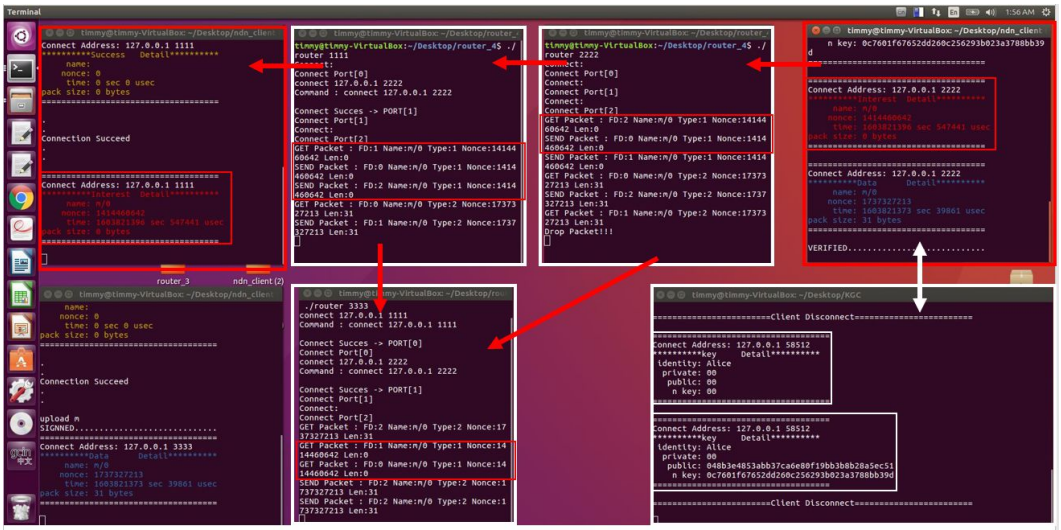
檔案生產者將要上傳的檔案與名稱合起來做雜湊並簽名、將簽章附在資料封包上、再上傳至 NDN 網路中與自己相鄰的路由器。如圖十三，Producer C 將檔案 m 做雜湊並簽名形成簽章，再將附有簽章與檔案 m 的資料封包（圖中藍色細框）傳至相鄰的 Router 3。



圖十三（將資料封包上傳至網路）

4.3 檔案要求

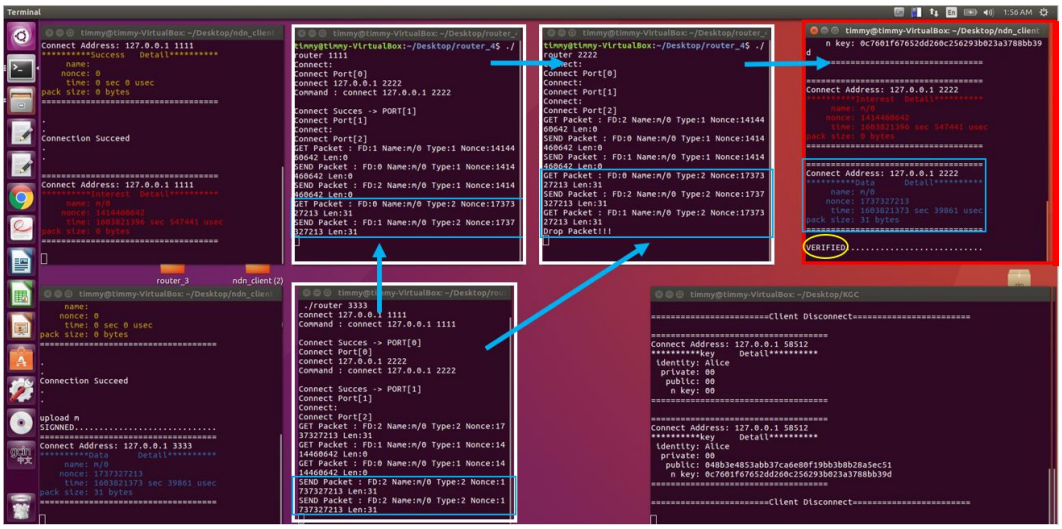
檔案要求者對相鄰的路由器送出要求檔案的興趣封包。網路中的路由器若無暫存對應的資料封包，會將興趣封包廣播至相鄰的路由器或使用者，直到暫存著對應資料封包的路由器，除此之外，若路由器收到了多個相同來源且相同要求的興趣封包，只會處理第一個到達的封包，並忽視其他較晚到達的封包。如圖十四，Consumer B 送出要求 m 的興趣封包（圖中的紅色細框），並以紅色的箭頭的方向將封包廣播，直到暫存著內容為 m 的資料封包的 Router 3，同時，連接至 KGC 並索取 Producer C 的公鑰，如白色箭頭所示，以用來認證未來收到的資料封包的來源。



圖十四（檔案要求者送出興趣封包）

4.4 檔案轉發

暫存著對應資料封包（圖中的藍色細框）的路由器將其依原路徑回傳至檔案要求者。如圖十五，Router 3 暫存著內容為檔案 m 的資料封包，因此將其依藍色箭頭的方向，以原路徑回傳至 Consumer B。



圖十五（路由器轉發資料封包）

4.5 驗證檔案合法來源

檔案要求者向 KGC 索取檔案生產者的公鑰，以此來認證封包中的簽章確認封包的來源。如圖十五，Consumer B 利用已取得的公鑰認證資料封包的來源。

章節 5
結論

在這次的專題中，我們使用 Unix socket 來模擬命名資料網路的架構，在這架構中，路由器的角色很類似我們生活中的伺服器，負責服務網路上的使用者，而使用者的角色類似客戶端，向路由器上傳資料或者請求資料。

在傳輸資料的過程中，為了避免使用者收到仿造的同名資料，我們實作了數位簽章，利用 MD5 和 RSA 來執行數位簽章，同時我們也製作了密鑰生成中心（Key Generation Center）給予使用者可信任的公鑰。藉由數位簽章與金鑰生成中心來維護命名資料網路中使用者的資料傳輸安全性。

5.1 實驗結果

透過本次專題實作，我們也展示出 NDN 的精神與符合今日對網路的使用需求的特點，包含下列五項：

(1) 以維護使用者匿名性的方式上傳或取得封包：在本次專題製作，使用者只知道與自己相鄰路由器的 IP 位址，在收到封包時也只會知道它是從路由器轉發而來，不會知道檔案生產者的 IP 位址。

(2) 在封包上就完成了傳輸的安全性：我們將資料的名稱與內容合併後再做簽章，因此確保封包的內容無法被隨意竄改，也無法使用相同名稱但不同內容的封包進行仿造。

(3) 路由器的封包暫存：在路由器對資料封包進行暫存，不同於 IP 路由器在轉發完封包後就將其丟棄。

(4) 較合適的檔案取得路徑：因路由器對於封包的暫存，對於已經被要求過的封包，使用者可以更快速且用更合理的路徑的取得它。

(5) 封包命名不會有空間不足及耗盡問題：在本次專題製作，我們使用 32bytes 的空間去儲存封包的名稱，相較於 IPv4 和 IPv6，幾乎不太可能遇到封命名空間不足的問題。

在實作與討論的同時，我們在實作中也發現 NDN 架構設計上的缺漏，舉例來說：Named Data Networking 中所提的路由器處理資料

封包順序（PIT、CS）會導致檔案生產者想主動上傳檔案至網路時，資料封包因在 PIT 中沒有對應的興趣封包紀錄而遭到刪除，因此我們將資料封包的處理順序 PIT 以及 CS 對調，讓使用者可以隨意上傳資料封包至網路中。

5.2 工學院聯合專題競賽評審問答

在工學院聯合專題競賽中，評審團主要以下列三點進行提問：(1) 如何使用與配置路由器暫存區的空間，(2) NDN 封包命名空間與 IP 位址空間的差異，與(3) NDN 是否能運用在小型區域網路中。對於上述評審團的提問，我們在下方進行說明並提出我們的看法。

(1) 如何使用與配置路由器暫存區的空間：路由器的暫存區（CS）利用 LRU 對資料封包進行暫存，以保留較常被使用到的檔案及資訊。

(2) NDN 封包命名空間與 IP 位址空間的差異：本次專題中我們將 NDN 封包的命名空間（Name 欄位）訂為 32 bytes，相比於 IP 位址的 32 bits，更不容易遭遇空間不足之問題。

(3) NDN 是否能運用在小型區域網路中：我們將此提問分為兩種不同的情況以方便說明。

(3.1) 在區域網路中的資料傳輸：在小型網路中使用 NDN 是可行的，但效果有限。僅在要求此區域網路範圍內的資料時能發揮其效能，若要求的資料尚未暫存在此網路中的路由器，仍須轉發要求至網際網路中。

(3.2) 從區域網路到網際網路的資料傳輸：NDN 與 IP 之封包傳輸方式不同，兩者間的轉換有著許多問題，舉例來說：若小型網路中的路由器沒有暫存著對應的資料封包，如何將 NDN 網路中要求資料的興趣封包，經由區域網路與網際網路間的介面轉換為在 IP 網路中的資料要求。這是一個超出本次專題製作的問題，雖然無法對其進行完整的解答，但仍是一個值得思考的發展方向。

5.3 未來展望

在這個科技發展快速的時代， IP 的問題逐漸地顯現出來，雖然命名資料網路是還在研發中，有許多內容及細節尚未定型。但是相信在未來命名資料網路的發展下，其越來越完善的架構能完整的展現其

特色，除了給人們快速搜尋資料的能力，也為物聯網等將會普及的科技發展帶來很大的幫助。

參考文獻

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, 2009.

[2] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang and Beichuan Zhang, “Named Data Networking”, ACM SIGCOMM Computer Communication Review, Volume 44, Number 3, July 2014.

[3] Whitfield Diffie, Martin E. Hellman, “New Directions in Cryptography”, IEEE Transaction on Information Theory, IT-22(6):644-654, Nov. 1976.

[4] Rivest, R., Shamir, A., and Adleman, L. “A method for obtaining digital signatures and public-key cryptosystems”, Comm. ACM 21, 2(Feb. 1978), 120-126.

[5] Rivest R., “The MD5 message-digest algorithm”, IETF Network Working Group, RFC 1321, Apr. 1992.

[6] S. S. AI-Riyami and K. G. Paterson. “Certificateless public key cryptography”, In Proc. ASIACRYPT 2003, pages 452-473. Springer-Verlag, 2003. LNCS 2894.

[7] Pei-Shan Yang, “File Transfer Protocol with Producer Anonymity for Named Data Networking”, 2020.

[8] https://www.itdaan.com/tw/959f63517fa2b8404a741d192982c8c2