

在 ICN 中基於內容請求熱門度之快取利用設計

許政穆
國立嘉義大學
資訊工程學系
hsujm@mail.ncyu.edu.tw

洪祥恩
國立嘉義大學
資訊工程學系
xyzusa419@gmail.com

摘要

Information Centric Network (ICN) 為一種以內容為中心的網路傳輸架構，所有繞送內容會被快取在轉送節點內容儲存記憶體內，當有相同資訊請求時，便可立即回應請求內容。透過資訊內容快取於 ICN 內，能使使用者更快取得所請求內容，並能提升使用者的使用感受。但當記憶體空間快取滿載時，必須有暫存內容被犧牲捨棄處理，因此好的內容快取機制，能有效提升 ICN 上資料傳輸使用效率。本論文提出以內容請求熱門度作為快取暫存依據，依內容請求熱門度的相關計算評估，決定在 ICN 內網路節點對內容快取處理，以避免在整體請求率較高的資訊內容因快取滿載而被優先置換，而造成下次相同內容請求時，還須向原始資訊供應端請求資訊內容，便能有效減緩 ICN 整體網路的資訊傳輸流量。

關鍵詞：內容中心網路、命名前綴、命名內容、快取利用、內容熱門度

Abstract

Information Centric Network (ICN) is a content-centric transmission framework. In an ICN node, all content will be routed by its name and temporarily cached in the content store of the node. If the same content is requested from another user, the ICN node already caching the requested content will immediately respond and deliver the requested content to the user. Through content is cached in ICN, the user can quickly obtain the requested content, and can enhance the users of use experience. In fact, the cache memory space is fully used, some cached content will be swapped out. An ideal content caching mechanism is needed for improving the efficiency of content delivery

in ICN. In this paper, a cache utilization based content popularity is proposed to decide which content will be continually cached or immediately swapped out from the content store in a ICN node for avoiding to request the same cache contents from the original providers to waste the communicating traffic. By the way, it can effectively reduce traffic volume of information flow in ICN.

Keywords: Information Centric Network, Name Prefix, Named Data, Cache Utilization, Content Popularity

1. 前言

目前網路上的應用不斷發展變化，基於 TCP/IP 的網路架構也開始出現一些問題，目前網路架構所暴露出的問題主要有：低擴展性、低安全性、缺乏靈活性等。根據這些問題，世界各地的研究團隊紛紛提出未來網路架構來解決目前的困境。

隨著網路資訊科技的進步，網際網路的使用量也隨之大幅提升。現行 TCP/IP 網路著重於以主機對主機(host-to-host)的網路傳輸架構，不管是在網路擴展、使用彈性、安全上都顯示此一傳輸架構在未來大量多媒體各式資訊內容的網路使用環境下已不敷使用。為了因應此一情形，故以內容為中心的網路 (Information Centric Network, ICN) 架構隨之產生。

以內容為中心的網路架構全球有多種的形式來實作。儘管這些形式細節所著重面都不一致，不過，主要目的還是致力於提升網路效能。以下為較有名組織包括： Network of Information (NetInf)[1]、 Publish-Subscribe Internet Routing Paradigm (PSIRP)[2]、 Content-Centric Networking (CCN)[3]、以及 Data-Oriented Network Architecture

(DONA)[4]。

在 NETINF 中，來源端(Source)發佈內容物件 ContentObject 至網路需向名為 NRS(name resolution service)來註冊，而 NetInf 中的節點都持有內容物件(Content Object)，NRS 會將最佳的來源位置來進行傳送。在 PSIRP 中，NDO 一樣從來源端發佈至網路，請求者可向網路來發起訂閱，而這雙方達到內容匹配(match)則是靠著著名的 rendezvous system 機制所完成。在 CCN 中，內容物件則被發佈至各個節點上，而每個節點包含三個重要元素:FIB、PIT、Content Store 負責命名資料(Named Data)的路由繞送與節點內快取。在 DONA 中，NDO 從資料來源被發布至網路上，每個節點是被允許可讀取資料的且被定義為 RH(resolution handlers) 解析處理器，用戶請求數據塊，接收到數據包後，可通過檢查包中公鑰的散列值來判斷是否來自於對應的主體。

圖 1 為傳統 TCP/IP 網路與 ICN 網路的結構比較。兩者架構分層有些類似，但主要不同地方在於沙漏中央處[3]。ICN 將命名塊取代原有的 IP，從整體網路架構來看，為將所有數據塊作命名而非整體資料作命名。另外，ICN 架構中的所有節點都具有儲存功能。所有經過的節點數據都會被儲存以被其他節點來做緩存，同時可以縮短數據回應時間，也可以大量減少網路中的傳輸流量。

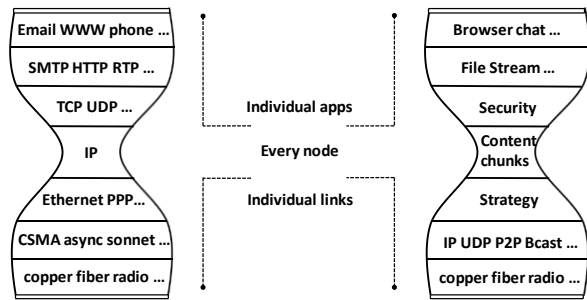


圖 1 TCP/IP 與 ICN 網路的架構比較

ICN 通訊動作為被動式的，也就是說是一個數據消費者。為了接收一個數據資料，請求端必須要發出一個興趣封包(Interest packet)，內容包含了所需資料名稱。舉例來說，當使用者請求了一個資料其命名為 com/discovery/tv-shows/moonshiners，路徑上的各個節點會記錄此興趣封包是由哪個傳輸介面(Interface)所傳進，接著藉由搜尋名稱將此興趣封包傳至 Forwarding Information Base (FIB)進行轉發動作。而當此興趣封包抵達擁有該內容的節點

後，並將命名資料以資料封包(Data Packet)封裝立刻回傳給資料請求者，而內容包含了內容本體及名稱，並且附上內容提供者所產生的簽章等資訊。也就是說資料封包會循著一開始興趣封包所繞送的路徑進行回傳直至請求端。由於興趣封包和資料封包都不包含任何主機位址或傳輸介面資訊，(如 IP 網路位址)，因此興趣封包是基於包含在內的內容名稱來進行遠送至內容提供者，而資料封包是基於興趣封包所繞送的路徑來進行回傳。

ICN 節點可能從下游接受多個重複興趣封包 ICN 節點會將興趣封包會集結成並往資料所在的上游節點來請求資料。此節點會將此資訊儲存至 Pending Interest Table (PIT)，PIT 內紀錄了各種興趣封包所進入的 Interface 及曾經符合興趣封包的 Interface。當資料封包抵達至該節點時，節點會找尋所有符合的 PIT 接口來做命名資料的回傳，接著將這些對應的 PIT 接口條目給刪除並且將命名資料快取至 Content Store (CS)，CS 基本上為節點路由器的快取記憶體用來儲存數據資料。

由於 ICN 網路節點快取資料封包於 ICN 網路內有很大的潛力，因它支援多種功能並且不需額外設備，包含了內容分發(content distribution)，如多個使用者在不同時間請求相同資料內容、群播(multicast)，如多個使用者在同一時間內請求相同資料內容、移動性(mobility)，如使用者在不同地區請求資料、以及延遲容忍網路(delay-tolerant networking)，如使用者所在網路具有間歇性連接情況發生)。例如，當使用者正在一個移動交通工具上觀看傳串流直播，請求端請求了一個資料內容並且移動到新地區網路，雖然請求資料會回傳至舊地點並且丟棄，但是請求資料已被快取至路徑上的所有節點內，故當請求端者再次請求此資料內容時，便會將最近的快取於網路內的資料內容回傳至請求端，使中斷時間達到最小化。當資料被快取至請求者附近節點可以提高封包傳送效率並且減少資料流的中途消失或攻擊。

由於傳統 ICN 必須將所有回傳的資料封包暫存下來，此一動作若是內容中含有多個較為冷門資料內容，將會使節點中的快取內容使用率大幅地降低，並且使得某些熱門內容產生額外的請求傳輸延遲。故本論文所提出一個依內容熱門度計算機制來評估 ICN 網路內快取內容的熱門程度，以用於可選擇將熱門內容給予優先保留，藉而提升節點中快取內容的利用率。使用此方法可以預期性的增加各節點中快

取記憶體命中率以及減少快取記憶體置換率。

2. ICN 快取機制

ICN 為一種新型的網路架構，在近幾年來已經有很多種不同方向的研究成果，不過在細節方面，例如快取策略還是有很大的研究空間，在傳統的 ICN 快取機制是一大特色，也是 ICN 能夠提高網路效率一個重要的手段之一，目前在 ICN 中的快取機制為 LRU(Least Recently Used) 或者 LFU(Least Frequently Used)，為最基本且最易符合長期用戶感興趣的替換機制，不過在於整體網路上用戶量高動態的變化此機制有時也會造成一些空間浪費。

為了解決此問題，本文提出了一種基於內容熱門度的機制來提供快取決策，可以有效為熱門內容提供較高的優先權來保留在各節點的記憶體中而不會被突來的冷門內容給替換掉。此演算法可以同時提高整體網路的內容使用率，也可減少整體網路的交通流量進而避免網路壅塞。

目前 ICN 中的 CS 最常使用的快取策略為 LRU(Least Recently Used) 或者 LFU(Least Frequently Used)。LRU(Least Recently Used)[6] 最久未使用演算法，它是根據歷史請求紀錄來進行選擇替換的數據，主要的概念為“如果內容近期被使用過，那麼未來也有較高的機率被使用”。相較於其他的快取機制來說，LRU 是較易來實現的，不過 LRU 並未考量到快取內容中的熱門度。例如，一個熱門內容很穩定的持續性地被請求，在接下來的時間有很高的機率會被再次請求，而根據 LRU 機制，如果在此時間內擁入了大量冷門新內容而此熱門內容將會被取代掉，就算此新冷門檔在未來有很少人請求還是會取代掉快取中的內容。

LFU(Least Frequently Used)[5] 為最近最少使用演算法，它是根據歷史請求頻率來進行選擇替換的數據，主要的概念為“如果內容近期被多次使用過，那麼未來也有較高的機率被使用”。當新的內容進入記憶體，且記憶體滿時，將會選擇次數最少的內容來進行取代。由於受歡迎的檔熱門內容將有較高的優先權保留在記憶體內，所以不會輕易的被取代掉，此種方法在靜態環境中具有相當好的表現，而在動態環境中是非常不適合的。例如，一內容在某時間內是相當受歡迎的所以此內容的計數次數

相當高，不過在接下來的一段時間即使沒有再次的請求此內容，此計數次數也不會下降，而只要接下來的內容請求率沒有高於它，此內容永遠都會保留在記憶體內不被取代掉。這樣的結果會導致一些之前熱門的內容，但最近很少使用的內容給佔據，使的新內容無法取代之。

經由以上快取機制的介紹及分析可以得出，為了提供整體網路的動態變化，快取機制必須要有適應快速變化的動態網路傳輸模式，不受突來的內容變化影響。有了以上的基本想法，接著本文將介紹將內容加入熱門度的計算以及取代機制。

3. 熱門度演算法

3.1 內容熱門度計算

在 ICN 中，使用熱門度來當作快取取代機制的演算法需要考量整體的請求量，且此熱門值之門檻將如何設置會在接下來的介紹中提及。

若是依照快取中所有請求內容的比例則會也不公平的現象，例如，節點 A 中擁有 2 個 a1 及 1 個 a2 的請求，節點 B 擁有 10 個 b1、100 個 b2 以及 90 個 b3 的請求。若是依照節點內請求內容的比例，節點 A 的 a1 將會佔有約 67% 的請求率，而節點 B 的 b3 佔有約 45%。在節點 B 中的 b3 很顯然的並非為最熱門之請求，不過請求數卻大於節點 A 的最熱門檔。單純根據節點內所有內容請求數比例會有冷門內容也成為熱門內容的現象。

在固定時間內利用請求的間隔時間來計算熱門排行，若是間隔分佈相當平均則在接下來的時間內將會有較高的機率來被存取到，所以熱門值相對較高。而要是在間隔時間內分佈相當零散則表示請求率相當不穩定，所以熱門值會對較低。利用此方法可以解決在不同節點有著不同請求數量比例的問題。對於單點的熱門度定義如下所示。

$$x = \frac{n}{\sum_{i=0}^n (t_{i+1} - t_i)^2} \dots \dots \dots (1)$$

t_i = request t_i 到達時間
 $t_0 = 0, t_{n+1}$ 欲計算時間
 n = request 次數

在式(1)中定義了節點內容的熱門值 \bar{x} ， t_i 表示請求封包 i 抵達的時間， t_0 為基準點預設為 0， t_{n+1} 預設為觀測間隔的值時間，n=內容請求次數。为了更好的說明均勻分布的請求封包會有較高的熱門值，以下將舉個 2 個案例來說明：

(1) 案例一：均勻分佈

假設觀測時間為 20 秒，Node1 以及 Node2 的請求封包圖如圖 2 所示。

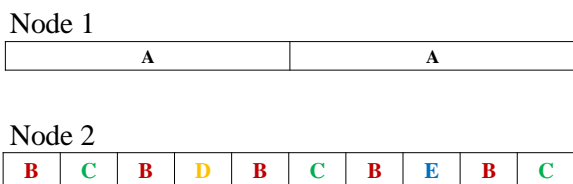


圖 2 案例一的請求封包數量

Node 1 的 content A 經計算後得到的值為： $2/(100+100)=0.01$ ，而 Node 2 之 content B 經計算後得到的值為： $5/(4+16+...+16+4)=0.069$ 。

由案例一可以發現 content A 不會因為單看請求封包比例而產生較高的熱門值。

(2) 案例二：零散分佈

假設觀測時間為 20 秒，Node1 以及 Node2 的請求封包圖如圖 3 所示。

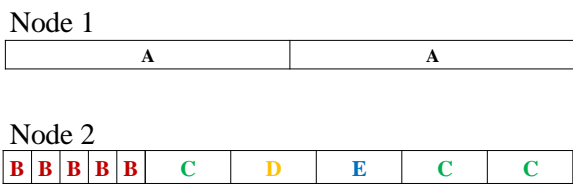


圖 3 案例二的請求封包數量

Node 1 之 content A 經計算後得到的值為： $2/(100+100)=0.01$ ，而 Node 2 之 content B 經計算後得到的值為： $5/(1+...+1+225)=0.021$ 。

由案例二可以發現均勻分布的熱門值會比較不均勻的熱門值要來的高，不過即使零散封包的熱門值很低也不會比 content A 來的低。

在定義完熱門度的計算方法後，接著要考

慮達到何種門檻值將會成為熱門檔。由式(1)的公式中可得知最差的情況為接近 $1/x^2$ ，x 為觀測時間，表示所有請求封包集中於開始或者結束的位置，而最佳熱門值為請求封包在觀測間隔時間內平均的分佈 $(n \times (n+1))/n^2$ 。

所以，故在節點內的內容熱門值範圍為如下所示。

$$\frac{1}{x^2} < \text{熱門值} \leq \frac{n \times (n+1)}{n^2}$$

3.2 內容熱門度之門檻值定義

依據式(1)單點熱門度定義公式可以得出此內容在此節點中的熱門值為多少，而超過多少值則將之定義為熱門度為此節所要討論的重點。

在觀測時間內，平均單位時間內至少接收到一筆請求封包則定義為熱門內容，在式(1)中，固定時間 x 內若有 n 個請求封包則最佳熱門度為 $(n \times (n+1))/n^2$ ，而此值會隨 n 增加而增大，藉由此方法控制可以將數量不足者進而篩選過濾掉，達到控制請求封包數量偏少不過卻是節點中最熱門的問題。

3.3 熱門度機制流程

在傳統的 ICN 中，只要繞送過的所有內容皆會儲存在本地的 Content Store 內，此一特色為 ICN 的一大優點不過在某方面來說也是一項缺點。即使持續性穩定地接收到某內容的請求，只要突來的大量冷門內容湧入節點將會使先前的熱門檔給取代之，這樣的結果將會造成內容請求延遲時間的增加。所以，在動態變化極大的網路中，使用傳統的 LRU 在某些情境下是不可行的。而以下將介紹使用熱門機制流程來提升整體 ICN 網路內容的使用率。

在一開始請求端的鄰近節點並沒有請求之內容，所以每個節點就必須逐漸的往內容上游端來做資料的請求。而內容提供者(Provider 端)一開始時在單位時間內將檢查是否有來自鄰居節點傳入的 info packet(包含了標註此內容為熱門以及此內容熱門值)以及經式(1)計算後得到 Content A 為熱門內容後接著將之標註為熱門內容。接著鄰居節點又有另一請求來請求 Content A,而此時鄰近節點所回傳的內容已標註為熱門,所以下游節點皆將本地 Content Store

內的 Content A 標註為熱門並且將所傳入的 interface 記錄至 FIB 中(表 1)。

表 1 FIB table

Prefix	Face List	Content Name
/parc.com	0, 1	Content A

一段時間後，由於鄰近節點已經有了 Content A 的副本，所以內容上游端 Content A 的請求量會自然的降低，而熱門度自然而然的也會降低。不過在同一時間內，下游端的某處會一直持續的收到 Content A 的請求，而經計算後 Content A 成為熱門內容後，除了回應所請求的 interface 之外，還會檢查 FIB 中的內容名稱(Content Name)欄位是否有無傳入過。若有，則將此 info packet 發送至對應的 interface，而鄰近節點接收到此 info packet 後會先檢查 Content Store 中是否擁有此內容，若有則標註為熱門檔並且檢查 FIB 的內容名稱(ContentName)欄位，而若此內容在此節點已經標註為熱門檔則丟棄此 info packet，若此節點中沒有此內容則直接檢查 FIB 中的內容名稱(ContentName)欄位。

回傳的 info packet 中內所包含的內容熱門值為當前熱門值遞減，而遞減目的是為了讓下游鄰居節點處於較高熱門狀態，擁有此內容的鄰居節點可逐漸減少熱門值來做優先替換動作。如此熱門內容在越靠近資料請求端將會有更高的熱門值來利用。

而此時網路拓樸中的節點只要皆收到此 info packet 後，在 CS (Content Store)中擁有此 Content A 皆會為熱門內容，且與節點本身所計算出的熱門檔為獨立事件。

4. 實驗及分析

在本文所提出的基於熱門度的快取機制有別於傳統的 LRU 機制，在以下實驗中將以基本實驗快取空間資料來驗證此演算法可行性。實驗環境中，拓樸將固定 Content Store 大小，也就是記憶體內物件堆疊大小，實驗節點中將由 100、300、500 種不同的物件種類所組成，並以隨機方式發送請求封包。對內容物的熱門值將預設以 Zip’f 定律[7]所分佈的值來定義，藉由固定快取記憶體大小來調整請求物件

的分散度，而實驗結果如圖 4 所示。實驗顯示當 LRU 在物件種類分散度提高時，其記憶體之擊中率將會逐漸降低，相較於本文所提出的基於熱門度的快取機制較為不穩定，而且基於熱門度之快取機制大至上都可保持約 40% 以上的命中率，可有效增加請求內容的搜尋速度，也可間接減少記憶體置換次數進而減少路由節點負擔。

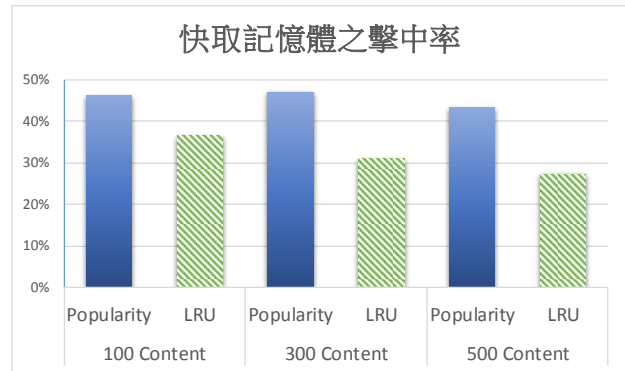


圖 4 快取記憶體之擊中率

對於後續實驗將會朝向進行擬真網路環境來建置並且實作本文提出的演算法來實際評估其實用性，以及透過評估結果進一步改進演算法設計。

5. 結論

ICN 為一種新型態的網路架構，其全新的路由方式由原本的“WHERE”轉變成為“WHAT”，一切致力於使網路不須內容儲存物的物理所在位置，而是直接提供連結內容的功能。ICN 的概念，已成為目前網路架構最熱門的研究方向。

本論文主要研究主題是將傳統 ICN 快取機制加上了一個熱門值來提供判斷取代優先權的條件，並且將 FIB 增加一欄位來記錄熱門內容所傳入的 interface，如此可以將熱門內容盡可能地保留在快取記憶體中以至下一個請求封包的到來，而非熱門的內容則會在快取記憶體中優先的來被取代掉，進而增加記憶體的利用率，這樣的方法在整體網路來看可以預期性地提高快取記憶體的擊中率以及減少記憶體的置換率，藉而提升網路數據的利用效能。

對於未來研究方向，將朝向以 NDN Project (<http://named-data.net/>) 與 CCNx (<http://www.ccnx.org/>) 作為本論文所題出以內

容熱門度為基礎的內容快取計算的實際實驗平台，以評估驗證本論文所提出的快取機制的效能，可作為 ICN 網路節點內容快取的實作參考。

致謝

本研究由國科會部份補助，計畫編號為 NSC 102-2221-E-415-017。

參考文獻

- [1] Ahlgren, B., Ambrosio, M.D., Dannewitz, C., Eriksson, A., Golić, J., Grönvall, B., Horne, D., Lindgren, A., Mämmelä, O., Marchisio, M., Mäkelä, J., Nechifor, S., Ohlman, B., Pentikousis, K., Randriamasy, S., Rautio, T., Renault, E., Seittenranta, P., Strandberg, O., Tarnauca, B., Vercellone, V., and Zeglache, D., “**Second NetInf Architecture Description**,” 4WARD EU FP7 Project, Deliverable D-6.2 v2.0, Apr. 2010, FP7-ICT-2007-1-216041- 4WARD / D-6.2, <http://www.4ward-project.eu/>.
- [2] Ain, M., Trossen, D., Nikander, P., Tarkoma, S., Visala, K., Rimey, K., Burbridge, T., Rajahalme, J., Tuononen, J., Jokela, P., Kjällman, J., Ylitalo, J., Riihijärvi, J., Gajic, B., Xylomenos, G., Savolainen, P., and Lagutin, D., “**D2.3 – Architecture Definition, Component Descriptions, and Requirements**,” Deliverable, PSIRP 7th FP EU-funded project, Feb. 2009.
- [3] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L., “**Networking named content**.” Proc. of the 5th international conference on emerging networking experiments and technologies, ser. CoNEXT '09.
- [4] Koponen, T., Chawla, M., Chun, B., Ermolinskiy, A., Hyun, K., Shenker, S., and Shenker, S., “**A Data-Oriented (and Beyond) Network Architecture**,” Proc. SIGCOMM '07, Kyoto, Japan, Aug. 27–31, 2007.
- [5] Lee, D., Choi, J., Kim, J.-H., Noh, S. H., Min, S. L., Cho, Y., and Kim, C. S., “**LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies**,” IEEE Trans. Computers, vol. 50, no. 12, pp. 1352–1360, 2001.
- [6] Megiddo, N., and Modha, D. S..

“**Outperforming lru with an adaptive replacement cache algorithm**.” IEEE Computer, 37(4):58–65, April 2004.

- [7] “**Zipf's**” ,[lawhttp://en.wikipedia.org/wiki/Zipf%27s_law](http://en.wikipedia.org/wiki/Zipf%27s_law)