

資料命名網路中以 CRC-32 為命名字首編碼設計

許政穆

國立嘉義大學
資訊工程學系

hsujm@mail.ncyu.edu.tw

張睿洋

國立嘉義大學
資訊工程學系

talkerr@gmail.com

摘要

資料命名網路(Named Data Networking, NDN)是一種以內容為主(content-centric)的傳輸網路，而非現今 TCP/IP 所使用主機位址定址(host-to-host)方式的傳輸網路，其注重是 WHAT(檔案內容)而非 WHERE(檔案位置)。資料命名網路中，內容提供者在資料發布時，會將資料儲存於 Content-Space (CS)中，並提供其唯一命名字首(named prefix)，當使用者對內容感興趣時，便以此字首在資料命名網路內搜尋取用。本論文所側重研究項目是改善以提高命名資料的搜尋效率，將每筆資料的唯一命名字首，以 CRC-32 (Cyclic Redundancy Check) 進行編碼，將命名字首以固定編碼長度取代原有儲存方式，編碼後不僅能將字首長度變短且讀取次數也減少許多，以提升了查詢效率。

關鍵詞：資料命名網路、命名前綴、命名檢索、循環冗餘校驗。

Abstract

Named Data Networking (NDN) is a content-centric transmission network. It is not similar to the traditional TCP/IP network which uses the IP addressing mechanism to deliver the network information. Thus, NDN focuses on WHAT (data content) instead of WHERE (data location). In NDN, a content provider publishes named data and stored them in its content storage, called Content-Space (CS). A published content has a unique name prefix. If a user wants to access the named content, the name prefix is used to lookup the corresponding content. In this paper, we adopt the encoding scheme, CRC-32 (Cyclic Redundancy Check) to encode the name prefix into a fix length encoded information for improving the efficiency of content lookup in NDN.

Keywords: Named Data Networking, Name Prefix Named Data Lookup, Cyclic Redundancy Check

1. 前言

隨著網際網路越來越發達，各種服務也如雪花般出現，這些服務需要透過網際網路來進行通訊，也代表將有各式各樣服務會在網路世界中被使用，而且其所需的流量會以驚人速度成長。目前雖然能夠透過不斷的研究與創新，來改良技術使網際網路更為完備，但假以時日這個問題只會越來越嚴重，這也顯示出目前這在使用的 TCP/IP 網路架構，已不滿足未來多樣化的資訊內容傳輸需求。故需要一個革命性的新架構來取代，而一個以內容為的網路架構，資料命名網路 (Named Data Networking, NDN)就此出現。

資料命名網路(Named Data Networking, NDN) [5]是一種新型態的網路架構，是由加州大學洛杉磯分校 Lixia Zhang 等人為首的團隊所開展的網路傳輸架構研究。資料命名網路是為了改變當前 TCP/IP 協定，以 content-to-content 的內容路由的資料傳輸繞送模式，而非 TCP/IP 協定的 host-to-host 的 IP 路由的資料繞送模式。

在資料命名網路中，所有節點都具有儲存功能，儲存的地方被稱為 Content-Space (CS)，CS 是以 B-TREE 結構來儲存資料，當 provider 提供將要發布的資料給節點的時候，節點就會讓該資料產生一串唯一的命名字首(Name Prefix)，並將資料放在 Content-Space 中。若當有使用者需提取資料時，只要利用該資料之 Name Prefix 作為資料搜索路由依據，亦即在 NDN 網路內搜尋到資料後，並沿著路徑回傳資料內容，在途中所有經過的節點資料都會被儲存以備其他節點來做提取，這樣可以大大減少網路中對於相同內容的請求存取流量。

為了改進在 NDN 網路中對資料內容以 Name Prefix 進行查尋效能，本論文將探討以 CRC-32 (Cyclic Redundancy Check)編碼機制對變動長度的 Name Prefix 進行固定長度編碼，讓 Name Prefix 查找的過程更為快速與簡便，

來提升搜尋速度，進而提高系統的效能。

本論文結構如下，第二節介紹相關背景研究，包含 NDN 以及 CRC 等議題，第三節介紹本論文所提出以 CRC-32 編碼的命名資料查找方法，第四節則為實驗結果以顯示 CRC-32 編碼的命名資料查找效果，最後第五節為結論。

2. 相關研究

2.1 資料命名網路 (Named Data Networking, NDN)

資料命名網路是由加州大學洛杉磯分校 Lixia Zhang 團隊為首開展的研究項目。資料命名網路和以內容為中心網路(Content Centric Networking, CCN)[3] 本質上幾乎沒有差別，而以內容為中心網路是由 PARC 的 Van Jacobson 在所提出。

資料命名網路的出現是為了改善當前已 TCP/IP 為主的 host-to-host，資料命名網路是 content-to-content，是以資料為主，根據資料給予唯一的 Prefix 當作名子而不是使用 IP，所以可以利用 Prefix 來搜尋資料。所以資料命名網路注重的是 WHAT(資料內容)而非 WHERE(資料位置)。

資料命名網路中的 Prefix 類似於現今網路中的 URL，不同的地方在於 URL 是有限長度的，Prefix 是無長度限制的。其格式是以 “/” 作為區分，例如: /parc/google/com/video.mpg 作為 video.mpg 的 Name Prefix。

資料命名網路有三個很重要結構:

- Content space (CS): CS 是儲存發布資料後的地方。CS 也會快取暫存其他節點轉送過來的資料。
- Pending Interest Table (PIT): PIT 是儲存所經過的 Interest Packet 資訊。因 Interest Packet 是透過節點來搜尋來源端所請求的命名資料，假若在搜尋該節點 CS 後，並未發現所請求資料，便將此 Interest Packet 請求內容記錄於 PIT 內，並再將 Interest Packet 轉發到其他節點，繼續資料請求。當 Interest Packet 所需資料在其他節點搜尋並回應資料內容，便會依據 PIT 資訊將請求內容繞送給請求者，並於 PIT 內刪除該筆 Interest Packet 請求資料的相關資訊。
- Forwarding Information Base (FIB): FIB

類似於 IP 路由器中的路由表，儲存各資料到下一個 Hop 的接口(Face)資訊。

資料命名網路使用的封包有兩種(圖 1):

- Interest Packet: 當請求者需要某項資料時便會使用含有該資料的 Prefix 的 Interest Packet 並轉發出去，其內容結構如圖 1[6]中的 Interest Packet 所示。
- Data Packet: 若該節點收到的 Interest Packet 的 Prefix 符合 CS 的資料，便會發送 Data Packet 經由沿途各節點 PIT 資訊返回請求者，其內容結構如圖 1 中的 Data Packet 所示。

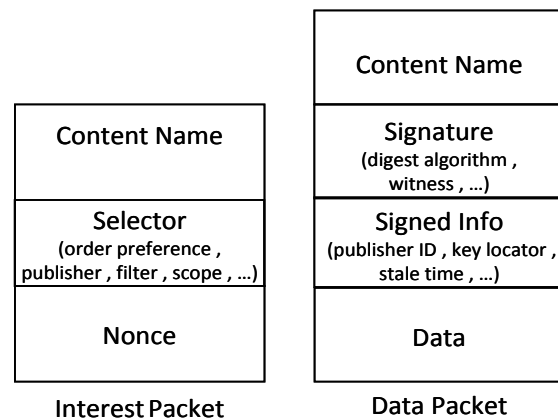


圖 1 封包格式

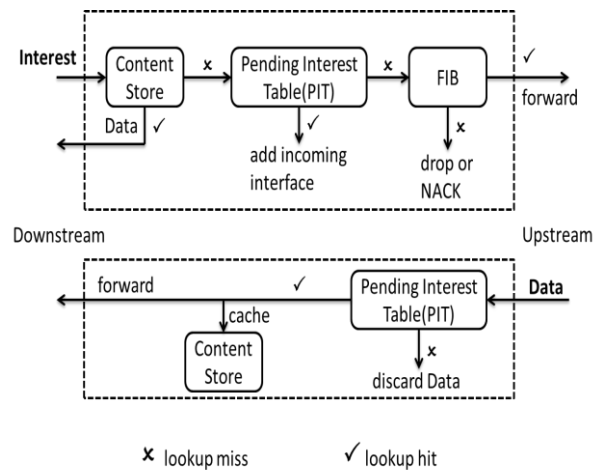


圖 2 NDN 命名資料處理流程

圖 2 為 NDN 網路典型的 Interest Packet 和 Data Packet 資料流的轉送處理流程[1]。當轉送 Interest Packet 到某一節點時，會先搜尋 CS 中有與 Name Prefix 符合的資料，若有則發送資料透過 Data Packet 並透過沿途各點 PIT 資訊返回請求者，若無則會搜尋 PIT 中，是否曾經有

其他一樣的 Interest Packet 經過，若有則丟棄該 Interest Packet，若無則建立 PIT 資訊，並透過 FIB 轉發 Interest Packet。

2.2 循環冗餘校驗 (Cyclic redundancy check, CRC)

循環冗餘校驗是由 W. Wesley Peterson 在他 1961 年發表的論文中所提出的一種方法，是根據網路資料產生簡短固定位數的驗證碼的一種雜湊函數，主要用來檢測資料傳輸是否出現的錯誤。CRC-a 是基於有限域 GF(2) (即除以 2 的同餘) 的多項式。循環冗餘校驗有很多不同種類 CRC-8、CRC-16、CRC-32、CRC64 等。不同種類的 CRC 差異在於使用的函式不同，其內容如圖 3 所示。

名稱	多項式	表示法:正常或翻轉
CRC-1	$x + 1$	0x1 or 0x1
CRC-8-CCITT	$x^8 + x^7 + x^3 + x^2 + x + 1$	0x8D
CRC-8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$	0xD5
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$	0x1021 or 0x8408
CRC-16-IBM	$x^{16} + x^{15} + x^2 + 1$	0x8005 or 0xA001
CRC-32-IEEE802.3	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	0x04C11DB7 or 0xEDB88320
CRC-64-ISO	$x^{64} + x^4 + x^3 + x + 1$	0x0000000000000001B OR 0xD800000000000000
...		
等		

圖 3 各種 CRC 函式表

對於 CRC 處理可以式(1) 的 M(x)表示，其中 X^n 為資料數據需左移 n 位，n 為資料數據表示為 2 進位的位數減 1，Q(x)為商，K(x)為欲要使用的 CRC 多項式，R(x)為校驗和，這通常用來檢查傳送後的資料是否正確，不過我們並不是拿來當作驗證的工具，而是當成類似雜湊函數，所以 R(x)也是我們所需要的結果。圖 4 為以 $x^4 + x^3 + x^2 + x + 1$ 使用 CRC 函式為 $x + 1$ ，處理結果為 1 之範例。

$$M(x) \cdot x^n = Q(x) \cdot K(x) - R(x) \dots\dots(1)$$

3. CRC-32 的命名字首編碼機制

本論文提出在 NDN 網路中的 Name Prefix 以 CRC 進行固定長度編碼，編碼過後不僅能

夠有效地縮短 Name Prefix 長度，也能減少記憶體使用量，並且加快搜尋速度，故 NDN 網路中對於資料流處理便如圖 5 所示。

目前選定的 CRC 是以 CRC-32 為主，因其編碼過後長度為 4 bytes。相對 CRC-16 的 2 bytes 跟 CRC-8 的 1 bytes 而言，長度越短也會有另外一個問題浮現，會造成資料查找碰撞問題。故以 CRC-32 的編碼後固定長度以足夠應付 NDN 網路內的 Name Prefix 編碼需求。

m0	1	1	1	1	1	0	0	0	0
p	1	1							
m1	0	0	1	1					
p			1	1					
m2		0	0	1	0				
p				1	1				
m3				0	1	0			
p					1	1			
m4					0	1	0		
p						1	1		
m5						0	1	0	
p							1	1	
R									1

圖 4 CRC 編碼處理示意

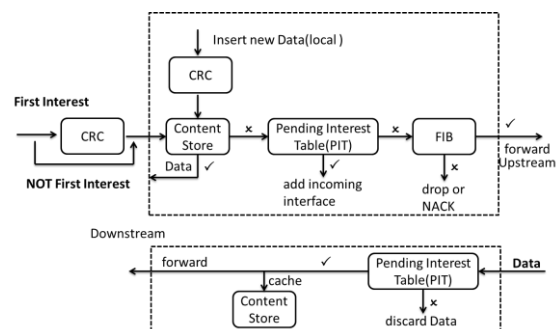


圖 5 加入 CRC Name Prefix 編碼的 NDN 命名資料處理流程圖

3.1 CRC-32 編碼方式

本論文採用 CRC-32-IEEE802.3 的 $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ 作為編碼，其 16 進位表示為

0x04C11DB7，但在使用時會反轉成 0xEDB88320 來使用。以 /parc/Referencia/Diccionarios/TerminoloX/Traballos_termino/index.html 為例，經過 CRC-32 編碼過後結果變成為 /CADCF501/8F4F1008/CCA53147/D65FFDAE/944232CB/774BE9C9，編碼前 Name Prefix 長度為 68bytes，當編碼後使用 28 bytes(可將 '/' 省略，每 8 個 bytes 為一段命名分節)，編碼後節省了約 58.82% 的空間，隨著 Name Prefix 越長節省的空间會越可觀。

以 Name Prefix 結構來看，其以 '/' 作為命名分節之用，若以包含 '/' 的 Name Prefix 進行 CRC-32 編碼，將無法清楚區隔命名分節，也無法對應到 NDN 節點 CS 中 B-Tree 結構，則會變成與原先一樣的線性搜尋處理效能，也會造成碰撞率會提升。若以搜尋角度來看，原有的 Name Prefix 唯一字串結構，而編碼後則成為數字結構，比較兩個讀取次數就會便會有顯著差異。以 Referencia 為例，編碼前需讀取比對 10 次，編碼後因為是數字所以只需要比對 1 次，故會提升搜尋 CS 效率。

3.2 False Positive

False Positive 意思是說，當有兩個字串編碼後得到相同結果，進而讀取到錯誤的資料。False Positive 會發生在所有使用過雜湊函式的資料上，同樣 CRC-32 也有可能發生此一狀況。

Genova 與 Christensen [2]，證明兩個 URL 經過 CRC-32 後碰撞的機率，因為 Prefix 類似於 URL，故本論文將採用 Genova 與 Christensen 所設計的式(2)與式(3)進行碰撞機率評估，其中式(2)是適用於不是由多個部件組成的 Prefix(如/www.google.com)，而式(3)是適用於 Prefix 是由多個部件組成。

$$P_{collision} = 1 - \left(\frac{2^k - 1}{2^K} \right)^{N-1} \dots\dots\dots(2)$$

式(3)中，k=32，其中 i 為 Prefix 有幾個部件組成，而 P_i 是 i 部件在整個 Prefix 的比例，I_{max} 為在所有 N 個 Prefix 中，最大的部件數量。

$$P_{collision} = \sum_{i=1}^{I_{max}} \left(1 - \left(\frac{2^k - 1}{2^K} \right)^{P_i^{N-1}} \right)^i \dots\dots\dots(3)$$

對於評估計算碰撞機率，將於第 4 節的實驗評估中加以說明。

4. 實驗評估

本實驗所使用的命名字首是以 DMOZ[4] 為資料集，於 2013 年 8 月共蒐集 68478 筆，隨機取 1000 筆作為實驗命名字首。

為了估算數據我們將會以式(4)與式(5)作為估算搜尋的數學方法。

$$\sum_{i=1}^c x_i \dots\dots\dots(4)$$

式(4)中的 i 為目前的部件，x_i 為部件的讀取次數，C 為同一 level 中部件的數量。故式(4)表示在 B-Tree 的同一 Level 中，從第一個部件到讀到正確部件所需的讀取次數。

$$\sum_{l=1}^n \sum_{i=1}^c x_i \dots\dots\dots(5)$$

式(5)為清單中 Prefix 的讀取次數的加總，l 為目前 Level，n 為部件總數。

4.1 讀取次數

實驗將從 DMOZ Dataset 中隨機選取的 1000 筆資料建成 B-Tree，並從 1000 筆中隨機選擇 100 筆作為資料搜尋測試，並且利用式(5)評估搜尋效率是否提升。

從 100 筆的命名字首查找實驗結果，經過處理後得到如表 1 數據，資料編碼後跟原資料做比較記憶體空間節省 29%，平均資料長度也減少了 65%。

表 1 實驗結果比較

	資料總容量	平均的資料長度	平均區段數	總讀取次數	平均讀取次數
NDN	38KB	159	6	47135	471
CRC-32-NDN	27KB	55	6	2418	24

表 1 中的讀取次數若要搜尋一筆資料，電腦在讀取資料期間的查詢次數，我們隨機選取 100 筆資料做搜尋，根據公式估算出的結果總讀取次數跟平均讀取次數都減少將近 19 倍，另外資料如果長度如果越長，則提升的效率會更顯著。

4.2 False Positive 機率

根據式(3)，因為使用 CRC-32，故 K 設為 32，對大部件數 I_{\max} 值設為 8，N 為 68478，計算出編碼後碰撞機率約為 0.00020%，機率微乎其微，且在計算讀取次數時也未碰到碰撞的情況，目前此機率很低，不足以發生碰撞，故先不考慮此一情況。

5. 結論

本論文使用 CRC-32 對資料命名往路中的資料命名進行固定長度編碼取代原先變動長度的資料命名字首(Name Prefix)，依據實驗結果估算約可提升效率為原來 19 倍之多。假若資料的命名字首越長，編碼後資料儲存記憶體就會省越多，也能提升資料命名字首在查找的搜尋效率。未來也將在本論文所提出的 CRC-32 命名字首編碼機制套用於 CCNx[3] 平台上，以驗證利用命名字首編碼機制可加快命名內容搜尋效率。

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] Bin L. , Huichen D., Junchen J., Keqiang H., Yan C. and Yi W.,” Scalable Name Lookup in NDN Using Effective Name Component Encoding”, **2012 IEEE 32nd International Conference in Distributed Computing Systems (ICDCS)**, pp. 668-697, 2012.
- [2] Christensen K. and Genova Z., “Managing routing tables for URL routers in content distribution networks,” **International Journal of Network Management**, Vol.14, pp.177-192, March 2004.
- [3] ”CCNx Project”, <http://www.ccnx.org>.
- [4] “DMOZ “, www.dmoz.com
- [5] Estrin D., Jacobson V., Zhang B. and Zhang L., “Named Data Networking Project,” Available: <http://www.named-data.net/>, NDN-0001, 2010.
- [6] Jihoon A., Young-Hee L. and Younggi K.,”CCNFRR: Fast one-hop Re-Route in CCN” , **2012 IEEE International Conference on Communications (ICC)**, pp. 5799 – 5803, 2012.