

在行動通訊網路下利用網路編碼執行群播重傳的排程機制

謝怡如

國立台中教育大學資工系
lucy80225@gmail.com

王讚彬

國立台中教育大學資工系
tpwang@mail.ntcu.edu.tw

Abstract

近年來行動通訊網路隨著智慧型手機的普及，使用率也迅速的上升。雖然行動通訊網路在移動性或是延展性都較傳統網路為佳以及方便，但是封包傳輸時可能會因為無線訊號的衰減或是訊號干擾而造成封包損毀或是封包遺失的情況發生，有時必須重傳封包，而封包重傳會增加整體網路的 overhead 造成傳輸成本的增加，因此將網路編碼(Network Coding)的技術帶入重傳機制中，進而減少重傳造成額外的 overhead。在此篇論文中，我們將提出一個改良式重傳排程演算法 Improved Hamming-Distance Algorithm，此演算法透過新增的條件篩選出較佳的封包合併組合，藉以提升吞吐量(throughput)，並降低被延遲的封包數量。

I. Introduction

現今社會中因智慧型手機使用率大幅的增加，行動通訊網路(mobile communication networks)的使用率也隨之增加。在行動通訊網路中，多媒體資料較常使用群播(multicasting)的方式來進行資料傳遞。而在行動通訊網路的網路模型中會有一個 base station (BS)透過無線訊號同時廣播資料(data)給所有在同一群組裡的所有行動終端節點(mobile terminal)。

雖然行動通訊網路在移動性或是延展性都較傳統網路為佳以及方便，但是封包傳輸時可能會因為無線訊號的衰減或是干擾而造成封包損毀或是封包遺失的情況發生。當發生封包損毀或是遺失時，行動終端節點會向 base station 發出一個封包重傳的請求。但是重傳會

增加 base station 的工作量並造成整體網路額外的頻寬負擔。為了要減少重傳帶來的 overhead 以及增進傳遞的吞吐量(throughput)，我們將把 Network Coding 的概念加入新提出的改良式封包重傳排程演算法。

Network Coding 的主要概念可以利用 butterfly example 來說明 Network Coding 是如何增進 throughput。參考 Fig. 1，假設當每條路徑每單位時間內的最大傳輸量皆為一個封包，S1 和 S2 要傳送 a_i 和 b_i 封包給 D1 和 D2。若我們利用中間的傳輸路徑傳輸 a_i 和 b_i 以 XOR 方式合併的 $a_i \oplus b_i$ 封包，則我們在單一單位時間內便可傳送兩個資料封包並且同時抵達目的節點，而目的節點因擁有 a_i 或 b_i 封包故可以順利解開 $a_i \oplus b_i$ 封包並得到所需要的封包。使用 Network Coding 的方式將原本需要四個單位時間才能傳遞完成的資料量縮短為三個單位時間便可達成，進而達到提升網路吞吐量。因此本篇論文將把 Network Coding 帶入封包重傳機制中，進而降低重傳對於整體網路的負擔，並且增加平均吞吐量以及減少封包延遲的時間。

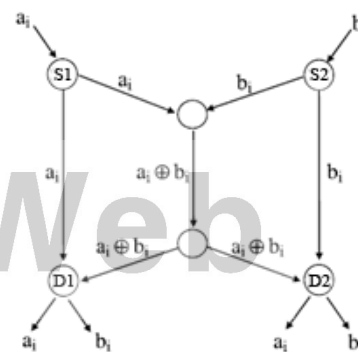


Fig. 1 Butterfly example for network coding

在此篇論文中，我們將改善 Hao WU *et al.* 所提出的 Hamming-Distance Algorithm [2]，考慮 Network Coding 成功解開 XOR 封包的機率以及封包合併的選擇性後，提出一個改良式重傳排程演算法 Improved Hamming-Distance Algorithm。

在第二節裡將詳細描述 Improved Hamming-Distance Algorithm 中會使用到的相關技術，包含 m -parallel stop-and-wait protocol 以及 Hamming-Distance Algorithm；第三節將詳細描述 Improved Hamming-Distance Algorithm 的運作方法；第四節將比較 Hamming-Distance Algorithm 以及 Improved Hamming-Distance Algorithm 兩種演算法的平均吞吐率以及平均 Delay 的封包數量。

II. Related Work

本節將詳細描述 Improved Hamming-Distance Algorithm 中使用到的重傳機制運作方式。2.1 節將介紹在 Hamming-Distance Algorithm 以及 Improved Hamming-Distance Algorithm 中所使用的重傳框架 m -parallel stop-and-wait protocol；2.2 節將介紹本論文所提出 Improved Hamming-Distance Algorithm 的原始演算法 Hamming-Distance Algorithm。

2.1 m -parallel stop-and-wait protocol

Improved Hamming-Distance Algorithm 將使用 m -parallel stop-and-wait protocol 為基礎，進行重傳的運作，而 m -parallel stop-and-wait protocol 的基礎原理是根據 stop-and-wait protocol 的方法進行。 m 代表 window size 並且 m transmission slots 構成一個 transmission cycle。

假設封包在第 l 回合 ($l=1, 2, 3 \dots$) 的 transmission slot 中被提出要求重傳，則此封包會在第 $l+m$ 回合的 transmission slot 被重傳。若無重傳要求時，則第 $l+m$ 回合的 transmission

slot 將被用來傳輸新的封包。

2.2 Hamming-Distance Algorithm [2]

另一方面，為降低重傳對於整體網路效能的影響，Hamming-Distance Algorithm 是以 Network Coding 為基礎的一種封包重傳排程機制。為了使 Hamming-Distance Algorithm 能順利的運作，我們需要在 BS 建立一個 Packet Status Table (如 TABLE 1)，用來儲存封包是否成功傳輸；其中每列代表各個行動終端節點，而每行代表一個 transmission slot 傳輸的所有封包。每個行動終端節點必須透過 NACK message 傳送自己的 user ID 和封包接收狀態，狀態 0 代表未收到封包，狀態 1 代表成功傳輸，而 Packet Status Table 也會依據此 NACK message 來更新狀態。

TABLE 1 Packet Status Table

Status:1 成功傳送封包；Status:0 封包需要重傳

	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆
U ₁	1	0	1	0	0	1
U ₂	1	1	1	1	0	0
U ₃	0	1	0	1	0	1
U ₄	0	0	1	0	1	1
U ₅	1	1	0	1	0	0
U ₆	1	1	0	0	1	0

Hamming-Distance Algorithm 是根據 Packet Status Table 中任兩行的 Hamming Distance 來決定 Network Coding 需要如何合併封包會有較好的結果。BS 將選擇 Hamming Distance 差異最大的兩個封包先做 Network Coding，因為 Hamming Distance 越大代表終端節點能順利解開此 XOR 封包的機率越大。而因為使用了

Network Coding 合併了重傳的封包，故新的 transmission slot 可以傳送比未使用 Network Coding 的 transmission slot 還要多個新的封包，進而提升網路整體的吞吐量，同時也降低了重傳的 overhead。

III. Improved Hamming-Distance Algorithm

在此篇論文中，我們提出的 Improved Hamming-Distance Algorithm 改善了 Hamming-Distance Algorithm [2] 對於相同 Hamming Distance 時選擇封包合併的條件，除了考慮 Hamming Distance 做為選擇封包合併的依據之外，在新提出的重傳排程演算法 Improved Hamming-Distance Algorithm 中多考量了下列兩個項目：(1) 合併後的 XOR 重傳封包是否能在此 transmission slot 中，讓所有終端節點皆成功收到原始的兩個封包 (2) 每個封包的合併選擇性。

Improved Hamming-Distance Algorithm 和 Hamming-Distance Algorithm 相同，也需要在 BS 建立一個 Packet State Table，用來儲存封包是否成功傳輸。選擇合併封包時，如 Fig. 2 演算法所示，首先考量任兩個封包對所有終端節點之 Hamming Distance 的大小，Hamming Distance 越大者優先合併封包。故將擁有最大 Hamming Distance 之封包組合放進 HD queue 中，若有多個封包組合 Hamming Distance 皆相同時，即 HD queue 中有各種封包合併的選擇，則再計算在 Packet State Table 中同列不同行的兩個狀態皆為 0 的數量(以下稱 00 數量)，數量越小者優先合併封包，故將 00 數量最小者放進 N00 queue 中。00 數量代表的意義為，此兩封包合併後有多少終端節點無法順利解開此封包，所以當 00 數為 0 代表此兩封包合併之後，在不掉包的假設下每個終端節點皆能收到此兩封包，也代表此兩封包合併重傳後，下一個 transmission slot 一定能傳輸新的封包，

因此挑選 00 數量越小者優先做 Network Coding。

```

Generate the transmission status matrix S and
count the Hamming Distance of any two packets;
/*建立 Packet State Table S 並計算 Hamming
Distance*/
Select the packet set which has the largest
Hamming Distance and put in into HD queue;
/*HD queue 用來儲存目前最大 Hamming
Distance 的封包組合*/
if HD queue has only one packet combination
Perform Network Coding to encode the packet
combination;
else {
Count the numbers of 00 number for each packet
combination in HD queue;
Select the packet combination with the smallest
00 number and put it into N00 queue;
/*N00 queue 用來儲存目前最小 00 數量的封包
組合*/
if N00 queue has only one packet
combination
Perform Network Coding to encode the packet
combination;
else
Select the packet combination which has the
smallest opportunity and perform Network
Coding to encode the packet combination; }

```

Fig. 2 Improved Hamming-Distance Algorithm

若 Hamming Distance 以及 00 數量皆相同時，即 HD queue 和 N00 queue 中皆有一個以上的封包合併選擇時，再比較這些合併組合中單一封包被選擇的機率。以 TABLE 2 為例， $P_1 \oplus P_6$ 、 $P_2 \oplus P_3$ 、 $P_2 \oplus P_6$ 此三個組合的 Hamming Distance 皆為 5 並且 00 數量皆為 0，會發現

P_1 以及 P_3 能選擇的合併封包數量較少，故此兩者優先考量，於是此回合的 Network Coding 結果為 $P_1 \oplus P_6$ 、 $P_2 \oplus P_3$ 、 $P_4 \oplus P_5$ 。

而在同一 transmission slot 中剩下的 window 將用來傳送新的封包。並每個行動終端節點必須透過 NACK message 傳送自己的 user ID 和封包接收狀態，而 Packet Status Table 會依據此 NACK message 來更新狀態。

TABLE 2 Coding Opportunity Table
(Hamming Distance / 00 number)

	P_1	P_2	P_3	P_4	P_5	P_6
P_2	2/1	-	-	-	-	-
P_3	3/1	5/0	-	-	-	-
P_4	3/1	1/2	4/1	-	-	-
P_5	4/1	4/1	3/2	5/1	-	-
P_6	5/0	5/0	2/2	4/1	3/2	-

IV. Performance Analysis

本節我們將分析改良式重傳排程演算法，並以簡單的例子說明 Improved

Hamming-Distance Algorithm 比原始的 Hamming-D Algorithm 整體有較高的平均吞吐量以及較少的封包延遲。

以 TABLE 1 以及 TABLE 2 為例，TABLE 2 是儲存以 TABLE 1 計算出的 Hamming Distance 以及 00 數量，根據 TABLE 2 結果比較此兩種重傳排程演算法發現，Hamming-D Algorithm 之 Network Coding 結果會有 Best Case： $P_1 \oplus P_6$ 、 $P_2 \oplus P_3$ 、 $P_4 \oplus P_5$ ，以及 Worst Case： $P_1 \oplus P_3$ 、 $P_2 \oplus P_6$ 、 $P_4 \oplus P_5$ ，平均吞吐量：28；平均 Delay 封包數：3(吞吐量為 XOR 封包成功傳遞並能順利解開之數量；Delay 封包數為此回合之 transmission slot 結束後，依舊需要

繼續重傳的原始封包數量)；Improved Hamming-Distance Algorithm 的 Network Coding 結果為 $P_1 \oplus P_6$ 、 $P_2 \oplus P_3$ 、 $P_4 \oplus P_5$ ，吞吐量：30；Delay 封包數：2。可以看出 Improved Hamming-Distance Algorithm 較 Hamming-Distance Algorithm 有較好的吞吐量，Delay 的封包數量也有減少。

因此由上述的例子可以看出 Improved Hamming-Distance Algorithm 較原始的 Hamming-Distance Algorithm 有更好的平均吞吐量，平均封包延遲數量也較少，為了驗證 Improved Hamming-Distance Algorithm 確實有比原始的 Hamming-D Algorithm 有更好的表現，我們將以下列方式做 Hamming-Distance algorithm 以及 Improved Hamming-Distance Algorithm 的比較：(1)假設 window size = 6；行動終端節點數 = 6 個(2)假設 window size = 10；行動終端節點數 = 10 個。以上述兩種假設分別以 5 組及 15 組平均來做兩種演算法平均吞吐量以及平均 Delay 封包數量的比較。

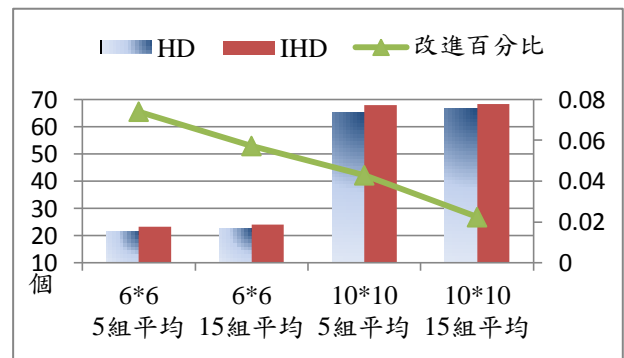


Fig. 3 成功傳遞封包數量

Fig. 3 表示成功傳遞的封包數量，假設在不掉包的情況下，能成功重傳並成功解開的封包數量。HD 代表 Hamming-Distance Algorithm；IHD 代表 Improved Hamming-Distance Algorithm；改進百分比為 $(IHD - HD) / HD * 100\%$ 。從 Fig. 3 可以看出 Improved Hamming-Distance Algorithm 比原始的

Hamming- Distance Algorithm 整體有較高的平均吞吐量，但隨著 window size 以及終端節點的增加，改善幅度下降。

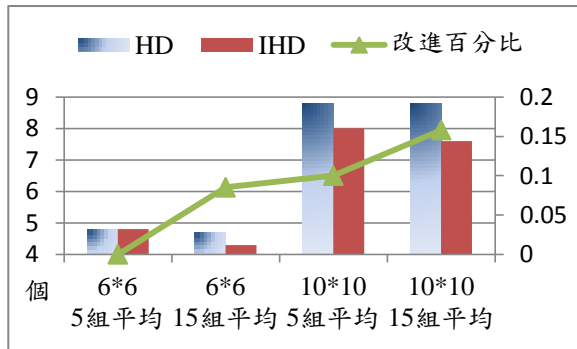


Fig. 4 Delay 封包數量

Fig. 4 表示 Delay 的封包數量，即重傳之後依舊在 transmission slot 中的封包數量。HD 代表 Hamming- Distance Algorithm；IHD 代表 Improved Hamming- Distance Algorithm；改進百分比為 $(HD - IHD)/HD * 100\%$ 。如圖所示，Hamming- Distance Algorithm 較 Improved Hamming- Distance Algorithm 有較多的封包產生 Delay 的狀況，因此 Hamming- Distance Algorithm 平均的封包延遲時間比 Improved Hamming- Distance Algorithm 還要長，並且隨著 window size 以及終端節點的增加，封包延遲的改善幅度也跟著上升。

透過 Fig. 3 及 Fig. 4 兩個測試案例可以發現，無論是平均吞吐量或是平均 Delay 封包數量，Improved Hamming- Distance Algorithm 都比原始的 Hamming- Distance Algorithm 有較好的吞吐量以及平均 Delay 的封包數量也比較少。

V. Conclusion

在行動通訊網路中，經常因為外在的干擾或是訊號的衰減，造成封包無法成功傳算，相較於一般網路環境，行動通訊網路會有較高的封包重傳成本。因此本論文所提出的改良式重傳排程演算法(Improved Hamming- Distance

Algorithm)是透過將 Network Coding 技術帶入重傳機制中，以增加重傳的吞吐量，並且可以減少整體網路的 overhead 以及減少封包 Delay 的時間。

Improved Hamming-Distance Algorithm 是改善 Hamming-Distance Algorithm 的一個新的重傳排程演算法。透過實際的模擬測試可以發現，無論是平均吞吐量或是平均 Delay 封包數量，Improved Hamming- Distance Algorithm 都比原始的 Hamming- Distance Algorithm 有較好的吞吐量，平均 Delay 的封包數量也較少。

致謝

This work was supported in part by the Ministry of Science and Technology under grant No. 102-2221-E-142-006-, Taiwan.

REFERENCE

- [1] Hao WU and Jun Zheng, "Network coding based multicast retransmission for mobile communication networks," Proceedings of Communication Technology (ICCT), 2010 12th IEEE International Conference, pp. 235-238, Nov 2010.
- [2] H. Wu and J. Zhen, "Efficient network coding-based multicast retransmission mechanism for mobile communication networks" Communications, IET, pp. 187 – 193, Jan 2012.
- [3] J. Zheng et al., "Network coding for wireless communication networks," IEEE Journal on Selected Areas in Communications, vol. 27, no. 5, pp. 577-581, June 2009.
- [4] N. Wang and N. Ansari, "Identifying the Network Coding Opportunity," Sarnoff Symposium, IEEE 2010.
- [5] Qiang Hu and J. Zheng, "An efficient

- network coding based multicast retransmission scheme for mobile communication networks with relays,” Communications (ICC), 2012 IEEE International Conference, Ottawa, ON, pp. 5413 - 5417, June. 2012.
- [6] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network information flow,” IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1204- 1216, July 2000.
- [7] S. Katti et al., “XORs in the air: practical wireless network coding,” IEEE/ACM Transaction on Networking, vol. 16, no. 3, pp. 497-510, June 2008.
- [8] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
- [9] Z. Dong, C. Zhan, and Y. Xu, “Delay aware broadcast scheduling in wireless networks using network coding,” in Proc. of the Second Int. Conf. on Network Security, Wireless Comm. and Trusted Computing, Hubei, Apr. 2010, pp. 214-217.

