

20. RFC 8163 : Transmission of IPv6 over Master-Slave/Token-Passing (MS/TP) Networks

RFC 8163 : 通過 MS / TP 網路傳輸 IPv6

網際網路工程任務組(IETF)

Request for Comments: 8163

分類: 標準

ISSN: 2070-1721

K. Lynn, Ed.

Verizon Labs

J. Martocci

Johnson Controls

C. Neilson

Delta Controls

S. Donaldson

Honeywell

2017 年 5 月

通過 MS / TP 網路傳輸 IPv6

摘要

主從/權杖傳遞 (MS / TP) 是用於 RS-485 實體層的媒體存取控制方法，主要用於建築自動化網路。該規範定義了用於傳輸 IPv6 分組的訊號框格式以及在 MS / TP 網路上形成鏈路本地和無狀態自動配置的 IPv6 位元元址的方法。

本文的狀態

這是一個網際網路標準協定文件。

本文是網際網路工程任務組的產品 (IETF)。它代表了 IETF 社群的共識。它有受到公眾審查，並已由網際網路工程指導小組 (IESG)。有關更多資訊 RFC 7841 的第 2 節提供了網際網路標準。

有關本文當前狀態的資訊，任何勘誤，以及有關如何提供反饋的資訊，請訪問

<http://www.rfc-editor.org/info/rfc8163>。

版權聲明

版權所有 (c) 2017 IETF Trust 和確定為文件作者的人員。版權所有。本文受 BCP 78 和 IETF Trust 關於 IETF 文件的法律規定 (<http://trustee.ietf.org/license-info>) 的約束，自本文發布之日起生效。請仔細閱讀這些文件，因為它們描述了您對本文的權利和限制。從本文中提取的代碼組件必須包含《信託法律條款》第 4.e 節中所述的 BSD 簡化許可證文本，並且如 BSD 簡化許可證中所述，不附帶任何保證。

目錄

1.	前言	3
2.	MS / TP上的IPv6的設定檔.....	6
3.	尋址方式	7
4.	最大傳輸單元 (MTU)	8
5.	LoBAC適應層	8
6.	無狀態位元元址自動配置.....	9
7.	IPv6鏈接本地地址	10
8.	單播位址映射	10
9.	多播位址映射	11
10.	標頭壓縮	11
11.	IANA 注意事項	12
12.	安全注意事項	12
13.	參考文獻	13
	附錄A. MAC介面摘要	16
	附錄B.恆定開銷的位元組填充[COBS].....	19
	附錄 C. Encoded CRC-32K [CRC32K].....	21
	附錄 D. 示例6 LoBAC訊號框解碼.....	23
	致謝	27
	作者資訊.....	28

1. 前言

主從/權杖傳遞 (MS / TP) 是用於 RS-485 [TIA-485-A] 實體層的媒體存取控制 (MAC) 協定，主要用於建築自動化網路。該規範定義了用於傳輸 IPv6 [RFC2460] 分組的訊號框格式以及在 MS / TP 網路上形成鏈路本地和無狀態自動配置的 IPv6 位元元址的方法。一般方法是使 6LoWPAN 規範 [RFC4944], [RFC6282] 和 [RFC6775] 的元素適用於受約束的有線網路，如下所述。

MS / TP 設備通常基於具有有限處理能力和記憶體的低成本微控制器。這些約束以及低資料速率和小 MAC 位元元址空間類似於 6LoWPAN 網路中面臨的限制。MS / TP 在至少三個方面與 6LoWPAN 顯著不同：a) MS / TP 設備通常由電源供電，b) 段上的所有 MS / TP 設備可以直接通信，因此沒有隱藏節點或網狀路由問題，並且 c) 最新的 MS / TP 規範為大型有效載荷提供支援，無需在 IPv6 下進行分段和重組。以下部分簡要概述了 MS / TP，然後介紹了如何在 MS / TP 訊號框中形成 IPv6 位址和封裝 IPv6 資料封包。該規範（隨後稱為“6LoBAC”）包括基於 LOWPAN_IPHC [RFC6282] 的 REQUIRED header 壓縮機制，並提高了 MS / TP 鏈路利用率。

1.1. 需求語言

本檔中的關鍵字“必須(MUST)”、“不得(MUST NOT)”、“必要(REQUIRED)”、“必須(SHALL)”、“不得(SHALL NOT)”、“應該(SHALL NOT)”、“不應該(SHOULD NOT)”、“建議(RECOMMENDED)”、“不建議(RECOMMENDED)”、“可以(MAY)”、“可選(OPTIONAL)”按照[RFC2119]中的描述進行解釋。

1.2. 被使用到的縮寫

ASHRAE：美國採暖，製冷和空調工程師協會
(<http://www.ashrae.org>)

BACnet：用於樓宇自動化和控制網路的 ISO / ANSI / ASHRAE 標準資料通信協定

CRC：循環冗餘碼

MAC：媒體訪問控制

MSDU：MAC 服務資料單元（MAC 客戶端資料）

MTU：最大傳輸單位；可以在單個網路事務中傳送的最大網路層協定資料單元的大小

UART：通用異步發送器/接收器

1.3. MS/TP 概覽

本節提供了 MS / TP 的簡要概述，如 ANSI / ASHRAE 標準 135-2016 [BACnet]第 9 章中所規定。[BACnet]的最新版本將對已提供的傳統 MS / TP(批准為[Addendum_an])的更改進行了整合，支援更大的訊號框大小和改進的錯誤處理。[BACnet]第 9 條還涉及實體層部署選項。

MS / TP 旨在通過遮罩雙絞線實現多點網路。它可以以 115.2 kbit / s 的資料速率支援長達 1000 米的網段，或者以較低的訊號框速率支援長達 1200 米的網段。MS / TP 介面僅需要 UART，帶有可禁用驅動器的 RS-485 [TIA-485-A]收發器和 5 ms 解析度計時器。MS / TP MAC 通常以軟體實現。

[TIA-485-A]使用的差分信號需要自由連接的 MAC。MS / TP 使用權杖來控制對多點匯流排的訪問。只有 MS / TP 主節點才能啟動未經請求的資料傳輸，並且只有在它持有權杖時才能啟動。在最多發送配置的最大資料訊號框數後，主節點將權杖傳遞給下一個主節點（由 MAC 位址確定）。如果鏈路上存在，則傳統的 MS / TP 實現（包括任何從節點）將忽略本規範中定義的訊號框格式。

[BACnet]第 9 節定義了一系列訊號框類型值，用於指定包含使用 Consistent Overhead Byte Stuffing [COBS]編碼的資料和資料 CRC 欄位的訊號框（參見附錄 B）。COBS 編碼的目的是從編碼資料和編碼 CRC-32K 欄位中消除前導序列。編碼資料由 32 訊號框 CRC [CRC32K](參見附錄 C)覆蓋，該 CRC 也是 COBS 編碼的。

MS / TP COBS 編碼訊號框具有以下格式：

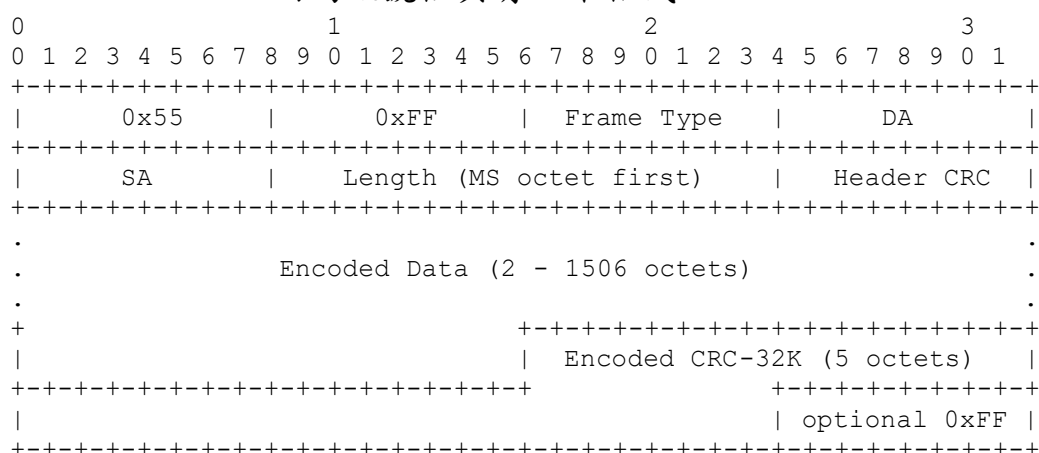


Figure 1: MS/TP COBS-Encoded 訊號框格式

MS / TP COBS 編碼的訊號框欄位定義如下：

前導碼	兩個訊號框組前導碼：0x55,0xFF
訊號框類型	一個訊號框組
目標地址	一個訊號框組地址
源地址	一個訊號框組地址
長度	兩個訊號框組，最重要的是訊號框組
標頭 CRC	一個訊號框組
編碼資料	2-1506 個訊號框組（參見第 4 節和附錄 B）
編碼 CRC-32K	五個訊號框組（見附錄 C）
(pad)	（可選）最多一個訊號框組的尾部：0xFF

訊號框類型用於區分不同類型的 MAC 訊號框。與此規範相關的類型（十進位）是：

- 0 Token
- 1 Poll For Master
- 2 Reply To Poll For Master
- 3 Test_Request
- 4 Test_Response
- ...
- 34 IPv6 over MS/TP (LoBAC) Encapsulation

訊號框類型 8 - 31 和 35 - 127 保留供 ASHRAE 分配。訊號框類型 32-127 指定 COBS 編碼的訊號框，其傳送編碼資料和編碼 CRC-32K 欄位。更多詳細資訊，請參見第 2 節。

目標和源位址的長度均為一個訊號框組。更多詳細資訊，請參見第 3 節。

對於 COBS 編碼的訊號框，長度欄位指示以八訊號框訊號框組為單位的[COBS]編碼資料欄位的大小再加上三個單位。（為了使傳統的 MS / TP 設備忽略 COBS 編碼的訊號框，需要進行此調整。）有關其他詳細資訊，請參閱第 4 節和附錄。

Header CRC 欄位包括訊號框類型，目標位址，源位址和長度欄位。Header CRC 生成和檢查程式在[BACnet]附件 G.1 中規定。

在[BACnet]第 9 章中討論了使用可選擇的 0xFF 尾部訊號框組。

1.4. 目標與限制

該規範的主要目標是：a) 通過最大程度地利用現有標準，直接在建築自動化和控制網路中的有線終端設備上啟用 IPv6，以及 b) 與傳統的 MS / TP 實現共存。共存允許 MS / TP 網路逐步升級以支援 IPv6。

為了與傳統設備共存，不允許對[BACnet]第 9 章中規定的 MS / TP 定址模式，訊號框 標頭格式，控制訊號框或主節點狀態機進行任何更改。

2. MS / TP 上的 IPv6 的設定檔

ASHRAE 已將 MS / TP 訊號框類型值指定為 34，以指示 IPv6 over MS / TP (LoBAC) 封裝。這屬於指定 COBS 編碼資料訊號框的值的範圍。

2.1. 強制性功能

[BACnet]第9條規定必須實施 MS / TP 設備的功能。例如，所有 MS / TP 節點必須用 Test_Response 訊號框回應 Test_Request。所有 MS / TP 主節點必須實現主節點狀態機並處理權杖，Poll For Master 和 Reply to Poll For Master 控制訊號框。6LoBAC 節點是 MS / TP 主節點，它是能夠實現處理 COBS 編碼訊號框的接收訊號框狀態機。

6LoBAC 節點必須支援 115.2 kbit / s 的資料速率，並且可以支援 [BACnet]第9章中規定的較低資料速率。選擇資料速率的方法超出了本規範的範圍。

2.2. 配置常量

接收訊號框狀態機使用以下常量。

Nmin_COBS_length 任何 LoBAC 封裝訊號框的最小有效長度值：
5

Nmax_COBS_lengt 任何 LoBAC 封裝訊號框的最大有效長度值：
1509

2.3. 配置參數

主節點狀態機使用以下參數。

Nmax_info_frames 節點在必須通過權杖之前可以發送的默認最大資訊訊號框數：1

Nmax_master 主節點的默認最高允許地址：127

設置參數或監控 MS / TP 性能的機制超出了本規範的範圍。

3. 尋址方式

MS / TP 節點 (MAC) 地址長度為一個八位元組並且地址是動態分配的。分配 MAC 位址的方法超出了本規範的範圍。但是，鏈

路上的每個 MS / TP 節點必須有唯一的位址，以確保正確的 MAC 操作。

[BACnet]第 9 節規定位址 0 到 127 對主節點有效。第 6 節中指定的用於創建 MAC 位址派生的介面識別符 (IID) 的方法可確保永遠不會生成所有零的 IID。

目標位址 255 (所有節點) 表示 MAC 層廣播。MS / TP 不支援組播，因此所有 IPv6 組播資料封包必須在 MAC 層廣播並在 IPv6 層進行過濾。絕不能使用 255 的源位址。

Hosts 根據[RFC4861]通過路由器通告學習 IPv6 前綴。

4. 最大傳輸單元 (MTU)

在傳輸時，網路層 MTU 根據第 5 節格式化並成為 MAC 服務資料單元 (MSDU)。然後 MSDU 由 MS / TP 進行 COBS 編碼。收到後，步驟相反。[BACnet]第 9 條支援長達 2032 個八位元組的 MSDU。

IPv6 [RFC2460]要求網際網路中的每個鏈路都具有 1280 位元組或更大的 MTU。此外，節點必須能夠接受分段資料封包，該資料封包在重新組裝後大到 1500 個八位元組。該規範定義了至少 1280 個八位元組和至多 1500 個八位元組的 MTU 長度。建議支援 MTU 長度為 1500 個八位元組。

5. LoBAC 適應層

本節規定了一個適配層，以支援第 10 節中規定的壓縮 IPv6 header。必須在所有節點上實現 IPv6 header 壓縮。實現也可以支援傳輸層標頭的通用標頭壓縮[RFC7400]。

本節定義的 LoBAC 封裝格式描述了 IPv6 over MS / TP 訊號框的 MSDU。LoBAC 有效載荷 (即，IPv6 分組) 遵循封裝 header 堆疊。LoBAC 是[RFC4944]中定義的 LoWPAN 封裝的子集，由 [RFC6282]更新，因此在下面的文字中使用“LOWPAN”是有意的。

LoWPAN和 LoBAC 封裝之間的主要區別在於後者中省略了 Mesh，Broadcast，Fragmentation 和 LOWPAN_HC1 頭。

通過 MS / TP 傳輸的所有 LoBAC 封裝資料報都以封裝頭堆疊為前綴，該封裝頭堆疊由 Dispatch 值後跟零個或多個頭欄位組成。當前為 LoBAC 定義的唯一序列是 LOWPAN_IPHC 頭，後跟有效負載，如下所示：

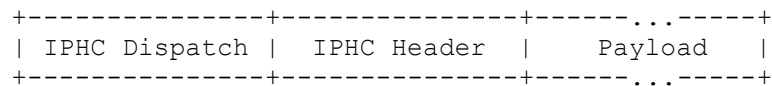


圖 2：一個 LoBAC 封裝的 LOWPAN_IPHC 壓縮的 IPv6 資料報

調度值被視為非結構化命名空間。只有一個模式用於表示當前的 LoBAC 功能。

Pattern	Header Type
01 1xxxxx	LOWPAN_IPHC - LOWPAN_IPHC compressed IPv6 [RFC6282]

圖 3：LoBAC 調度值位元元模式

除非另有說明，否則其他由 IANA 分配的 6LoWPAN 調度值不適用於 6LoBAC。

6. 無狀態位元元址自動配置

本節定義如何獲取 IPv6 介面識別符。該規範區分兩種類型的 IID，MAC 地址派生和語義不透明。

MAC 位址派生的 IID 是用於形成鏈路本地地址的 RECOMMENDED 類型，因為它提供了由第 10 節中規定的 LOWPAN_IPHC [RFC6282]格式提供的最有效的 header 壓縮。創建 MAC 的一般過程 - 地址衍生的 IID 在[RFC4291]附錄 A“創建修改的 EUI-64 格式介面識別符”中描述，由[RFC7136]更新。

鏈路本地地址的介面識別符應該通過將節點的 8 位 MS/TP MAC 地址連接到七個八位元組 0x00,0x00,0x00,0xFF, 0xFE, 0x00,0x00 來形成。例如，十六進制值 0x4F 的 MS/TP MAC 地址會產生以下 IID：

0	1	3	4	6
0	5	1	7	3
+-----+-----+-----+-----+				
0000000000000000 0000000011111111 1111111000000000 0000000001001111				
+-----+-----+-----+-----+				

對於每個全域範圍的位址，建議使用具有 64 位熵的語義不透明 IID，並且可以根據第 12 節中引用的方法之一在本地生成。生成 64 位語義不透明 IID 的節點必須使用其本地註冊 IID 路由器通過發送帶有地址註冊選項（ARO）的 Neighbor Solicitation（NS）消息並根據[RFC6775]處理 Neighbor Advertisements（NA）。

用於 MS / TP 介面的無狀態自動配置[RFC4862]的 IPv6 位址前綴必須具有 64 位的長度。

7. IPv6 鏈接本地地址

MS / TP 介面的 IPv6 本地鏈接位址[RFC4291]是通過將介面識別符（如上定義）附加到前綴 FE80 :: / 64 來形成的。

10 bits	54 bits	64 bits
+-----+-----+-----+		
1111111010 (zeros) Interface Identifier		
+-----+-----+-----+		

8. 單播位址映射

除非另有說明，否則將 IPv6 非多播位址映射到 MS / TP MAC 層位址的位址解析過程遵循[RFC4861]第 7.2 節中的一般描述。

當地址是 8 位 MS / TP MAC 層（節點）地址時，源/目標鏈路層位址選項具有以下形式。

0	1														
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5															
+-----+-----+-----+															
Type								Length=1							
+-----+-----+-----+															
0x00								MS/TP Address							
+-----+-----+-----+															
+ Padding (all zeros) +															
+-----+-----+-----+															

選項欄位：

類型：

- 1：用於源鏈路層地址。
- 2：用於目標鏈路層地址。

長度：這是此選項的長度（包括類型和長度欄位），以 8 個八位元組為單位。對於 8 位 MS / TP MAC 地址，該欄位的值為 1。

MS / TP 地址：規範位順序的 8 位地址[RFC2469]。這是介面當前響應的單播位址。

9. 多播位址映射

必須將所有 IPv6 多播資料封包發送到 MS / TP 目標地址 255（廣播）並在 IPv6 層進行過濾。當在壓縮 Header 中表示為 16 位位址時（參見第 10 節），它必須通過左邊的填充以零八位元組形成：

```

0                                     1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           0x00           |           0xFF           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

10. 標頭壓縮

6LoBAC 需要 LOWPAN_IPHC IPv6 壓縮，其在[RFC6282]中規定並且通過引用包含在此。本節將簡單地確定在解釋[RFC6282]的文本時應該進行的替換。

一般來說，應該進行以下替換：

- 用“MS / TP 網路”替換“6LoWPAN”的實例
- 用“MS / TP 地址”替換“IEEE 802.15.4 地址”的實例

當需要一個 16 訊號框位址（即一個 IEEE 802.15.4“短地址”）時，它必須通過用零訊號框組填充左邊的 MS / TP 位址來形成：

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      0x00      | MS/TP address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

如果 LOWPAN_IPHC 壓縮[RFC6282]與上下文一起使用，則直接連接到 MS / TP 段的路由器必須根據[RFC6775]，第 7.2 節傳播 6LoWPAN 上下文選項（6CO）。

11. IANA 注意事項

本文使用[RFC4944]和[RFC6282]之前保留的值，不再向 IANA 提出請求。

RFC Editor 的注意事項：本部分可能會在發佈時刪除。

12. 安全注意事項

有關受約束節點所面臨的隱私威脅的一般性討論，請參閱[RFC8065]。

[RFC8065]區分“穩定”和“臨時”地址。前者是長期存在的，通常由服務器做廣告。後者通常由客戶使用，並且應該經常更改，以減少活動隨時間的相關性。參與這兩項活動的節點應該支援每台設備同時使用多個位址。

包含 MAC 位址派生的 IID 的全域範圍位址可能會暴露網路以解決掃描攻擊。因此，建議根據例如[RFC3315]，[RFC3972]，[RFC4941]，[RFC5535]或[RFC7217]為每個使用的全域範圍位址生成 64 位語義不透明 IID。

13. 參考文獻

13.1. 規範性參考

[BACnet] American Society of Heating, Refrigerating, and Air Conditioning Engineers, "BACnet - A Data Communication Protocol for Building Automation and Control Networks", ANSI/ASHRAE Standard 135-2016, January 2016, <http://www.techstreet.com/ashrae/standards/ashrae-135-2016?product_id=1918140#jumps>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6(IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<http://www.rfc-editor.org/info/rfc3972>>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6

(IPv6)", RFC 4861, DOI 10.17487/RFC4861,
September 2007,
<<http://www.rfc-editor.org/info/rfc4861>>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6
Stateless Address Autoconfiguration",
RFC 4862, DOI 10.17487/RFC4862, September
2007,
<<http://www.rfc-editor.org/info/rfc4862>>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy
Extensions for Stateless Address
Autoconfiguration in IPv6", RFC 4941, DOI
10.17487/RFC4941, September 2007,
<<http://www.rfc-editor.org/info/rfc4941>>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D.
Culler, "Transmission of IPv6 Packets over IEEE
802.15.4 Networks", RFC 4944, DOI
10.17487/RFC4944, September 2007,
<<http://www.rfc-editor.org/info/rfc4944>>.

[RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC
5535, DOI 10.17487/RFC5535, June 2009,
<<http://www.rfc-editor.org/info/rfc5535>>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for
IPv6 Datagrams over IEEE 802.15.4-Based
Networks", RFC 6282, DOI 10.17487/RFC6282,
September 2011,
<<http://www.rfc-editor.org/info/rfc6282>>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C.
Bormann, "Neighbor Discovery Optimization for
IPv6 over Low-Power Wireless Personal
Area Networks (6LoWPANs)", RFC 6775, DOI
10.17487/RFC6775, November 2012,
<<http://www.rfc-editor.org/info/rfc6775>>.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6

Interface Identifiers", RFC 7136, DOI
10.17487/RFC7136, February 2014,
<<http://www.rfc-editor.org/info/rfc7136>>.

[RFC7217] Gont, F., "A Method for Generating Semantically
Opaque Interface Identifiers with IPv6 Stateless
Address Autoconfiguration (SLAAC)", RFC
7217, DOI 10.17487/RFC7217, April 2014,
<<http://www.rfc-editor.org/info/rfc7217>>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header
Compression for IPv6 over Low-Power Wireless
Personal Area Networks (6LoWPANs)",
RFC 7400, DOI 10.17487/RFC7400, November
2014,
<<http://www.rfc-editor.org/info/rfc7400>>.

13.2. 資訊參考

[Addendum_an] American Society of Heating, Refrigerating, and
Air Conditioning Engineers, "ANSI/ASHRAE
Addenda an, at, au, av, aw, ax, and az to
ANSI/ASHRAE Standard 135-2012,
BACnet - A Data Communication Protocol for
Building Automation and Control
Networks", July 2014,
<[https://www.ashrae.org/File%20Library/docLib/StdAddenda/
07-31-2014_135_2012_an_at_au_av_aw_ax_az
_Final.pdf](https://www.ashrae.org/File%20Library/docLib/StdAddenda/07-31-2014_135_2012_an_at_au_av_aw_ax_az_Final.pdf)>.

[COBS] Cheshire, S. and M. Baker, "Consistent Overhead Byte
Stuffing", IEEE/ACM TRANSACTIONS ON
NETWORKING, VOL.7, NO.2 , April 1999,
<<http://www.stuartcheshire.org/papers/COBSforToN.pdf>>.

[CRC32K] Koopman, P., "32-Bit Cyclic Redundancy Codes for
Internet Applications", IEEE/IFIP International
Conference on Dependable Systems and
Networks (DSN 2002), June 2002,
<https://users.ece.cmu.edu/~koopman/networks/dsn02/dsn02_koo

pman.pdf>.

[IEEE.802.3_2012] IEEE, "802.3-2012", IEEE 802.3-2012, DOI 10.1109/ieeestd.2012.6419735, January 2013, <<http://ieeexplore.ieee.org/servlet/opac?punumber=6419733>>.

[RFC2469] Narten, T. and C. Burton, "A Caution On The Canonical Ordering Of Link-Layer Addresses", RFC 2469, DOI 10.17487/RFC2469, December 1998, <<http://www.rfc-editor.org/info/rfc2469>>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <<http://www.rfc-editor.org/info/rfc8065>>.

[TIA-485-A] Telecommunications Industry Association, "TIA-485-A, Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (ANSI/TIA/EIA 485-A-98) (R2003)", March 2003, <https://global.ihs.com/doc_detail.cfm?item_s_key=00032964>.

本附錄內容訊息豐富，不屬於標準的一部分。

[BACnet]第 9 節通過其 SendFrame 和 ReceivedDataNoReply 過程為 MAC 層客戶端提供支援。但是，它沒有為 MAC 定義獨立於網路協定的抽象介面。這在下面提供，作為實施的輔助。

A.1. MA-DATA 請求

A.1.1. 功能

該原語定義了在廣播地址的情況下從 MAC 客戶端實體到單個對等實體或多個對等實體的資料傳輸。

A.1.2. 服務原始語句的語義

原始語義如下：

```
MA-DATA.request (  
    destination_address,  
    source_address,  
    data,  
    type  
)
```

'destination_address'參數可以指定個體或廣播 MAC 實體位址。它必須包含足夠的資訊來創建本地 MAC 子層實體在訊號框前面添加的目標地址欄位（參見第 1.3 節）。

'source_address'參數（如果存在）必須指定單獨的 MAC 地址。如果省略 source_address 參數，則本地 MAC 子層實體將插入與該實體相關聯的值。

'data'參數指定由 MAC 子層實體傳送的 MAC 服務資料單元（MSDU）。存在與 MAC 子層實體的 MSDU 相關聯的足夠信息以確定資料單元的長度。

'type'參數指定由本地 MAC 子層實體預先添加到訊號框的 MS / TP 訊號框類型欄位的值。

A.1.3. 何時生成

每當資料被傳送到對等實體時，該原語由 MAC 客戶端實體生成。這可以響應來自更高協定層的請求或來自內部生成到 MAC 客戶端的資料，例如權杖訊號框。

A.1.4. 對接收的影響

接收該原語將導致 MAC 實體插入所有 MAC 特定欄位，包括目標位址，源位址，Frame 類型以及特定媒體訪問方法唯一的任何欄位，並將正確形成的 Frame 傳遞給較低協定層，用於轉移到對等 MAC 子層實體。

A.2. MA-DATA.indication

A.2.1. 功能

該原語定義了在廣播地址的情況下從 MAC 子層實體到 MAC 客戶端實體的資料傳輸。

A.2.2. 服務原始語句的語義

原始語句如下：

```
MA-DATA.indication (  
    destination_address,  
    source_address,  
    data,  
    type  
)
```

'destination_address' 參數可以是由輸入訊號框的目標位址欄位指定的個人或廣播地址。 'source_address' 參數是由傳入訊號框的源地址欄位指定的單個地址。

'data' 參數指定本地 MAC 實體接收的 MAC 服務資料單元 (MSDU)。有足夠的信息與 MAC 子層客戶端的 MSDU 相關聯，以確定資料單元的長度。

'type'參數是傳入訊號框的 MS / TP 訊號框 Type 欄位的值。

A.2.3. 何時生成

MA_DATA.indication從 MAC 子層實體傳遞到 MAC 客戶端實體，以指示 Frame 到達目的地為 MAC 客戶端的本地 MAC 子層實體。僅當這些訊號框有效地形成，沒有錯誤地接收並且它們的目的地位址指定本地 MAC 實體時才報告這些訊號框。發往 MAC 控制子層的訊號框不會傳遞到 MAC 客戶端。

A.2.4. 對接收的影響

未指定 MAC 客戶端接收此原語的效果。

附錄 B. 恆定開銷的位元組填充[COBS]

本附錄內容訊息豐富，不屬於標準的一部分。

[BACnet]第 9 條糾正了 MS / TP 規範的長期問題;即，只要前導序列出現在資料或資料 CRC 欄位中，它們就不會被轉義。在極少數情況下，由於訊號框同步丟失，導致訊號框丟失。解決方案是在使用一致開銷位元組填充[COBS]進行傳輸之前對資料和 32 位資料 CRC 欄位進行編碼，並在接收時解碼這些欄位。

COBS 是一種運行長度編碼方法，名義上從其輸入中刪除“0x00”八位元組。可以通過用 COBS 輸出的每個八位元組對該值進行異或來消除任何選擇的八位元組值。[BACnet]第 9 條規定刪除前導八位元組'0x55'。

COBS 的最小開銷是每個編碼欄位一個八位元組。如[COBS]中所述，長欄位中的最壞情況開銷被限制為每 254 個一個八位元組。

訊號框編碼在邏輯上以兩遍進行。通過將 MSDU 傳遞通過 COBS 編碼器並對輸出的每個八位元組對前導碼八位元組'0x55'進行異或來準備編碼資料欄位。然後，通過在編碼資料欄位上計算 CRC-32K 並將其格式化以用於傳輸來準備編碼的 CRC-32K 欄位，如附錄 C 中所述。這些欄位的組合長度減去兩個八位元組以便與

傳統的 MS / TP 設備相容，在傳輸之前放置在 MS / TP 標頭長度欄位中。

COBS 編碼器和解碼器功能示例如下所示。[BACnet]附件 T 中提供了使用和測試向量的完整示例。

```
<CODE BEGINS>
#include <stddef.h>
#include <stdint.h>
/*
 * Encodes 'length' octets of data located at 'from' and
 * writes one or more COBS code blocks at 'to', removing any
 * 'mask' octets that may present be in the encoded data.
 * Returns the length of the encoded data.
 */

size_t
cobs_encode(uint8_t *to, const uint8_t *from, size_t length,
            uint8_t mask)
{
    size_t code_index = 0;
    size_t read_index = 0;
    size_t write_index = 1;
    uint8_t code = 1;
    uint8_t data, last_code;

    while (read_index < length) {
        data = from[read_index++];
        /*
         * 在資料中遇到非零位元組的情況下，只需將輸入複製到輸出並遞增代碼八位元組。
         */
        if (data != 0) {
            to[write_index++] = data ^ mask;
            code++;
            if (code != 255)
                continue;
        }
        /* 在遇到資料中的零或已複製非零八位元組的最大數量（254）的情況下，
         * 存儲代碼八位元組並重置編碼器狀態變量。
         */
        last_code = code;
        to[code_index] = code ^ mask;
        code_index = write_index++;
        code = 1;
    }

    /*
     * 如果最後一塊恰好包含 254 個非零八位元組，則此異常已在上面處理（返回的長度必須是已調整的）。否則，正常編碼最後一塊，就像在數據後附加“幻影零”。如果最後一個塊包含恰好 254 個非零八位元組，則上面處理它的異常（並且必須調整返回的長度）。否則，正常編碼最後一個塊，就像將“幻影零”附加到資料一樣。
     */
    if ((last_code == 255) && (code == 1))
```

```

        write_index--;
    else
        to[code_index] = code ^ mask;

    return write_index;
}

#include <stddef.h>
#include <stdint.h>
/*
 * Decodes 'length' octets of data located at 'from' and
 * writes the original client data at 'to', restoring any
 * 'mask' octets that may present in the encoded data.
 * Returns the length of the encoded data or zero if error.
 */
size_t
cobs_decode(uint8_t *to, const uint8_t *from, size_t length, uint8_t mask)
{
    size_t read_index = 0;
    size_t write_index = 0;
    uint8_t code, last_code;
    while (read_index < length)
    {
        code = from[read_index] ^ mask;
        last_code = code;
        /* * Sanity check the encoding to prevent the while() loop below
         * from overrunning the output buffer.
         */
        if (read_index + code > length)
            return 0;
        read_index++;
        while (--code > 0)
            to[write_index++] = from[read_index++] ^ mask;
        /*
         * Restore the implicit zero at the end of each decoded block * except when it contains exactly
         254 non-zero octets or the
         * end of data has been reached.
         */
        if ((last_code != 255) && (read_index < length)) to[write_index++] = 0;
    }
    return write_index; }

<CODE ENDS>

```

附錄 C. Encoded CRC-32K [CRC32K]

本附錄內容豐富，不屬於標準的一部分。

將 MS / TP 的有效載荷擴充到 1500 個八位元組需要將資料 CRC 從 16 位升級到 32 位。P.Koopman 撰寫了幾篇關於評估網路應用的 CRC 多項式的論文。在[CRC32K]中，他調查了整個 32 位多項式空間，並注意到一些超出[IEEE.802.3_2012]多項式的性能。[BACnet]第 9 條規定了其中之一，即 CRC-32K(考夫曼)多項式。

calc_crc32K () 函數的指定用法如下。在傳輸訊號框之前，'crc_value'被初始化為所有訊號框。在通過函數傳遞[COBS]編碼資料的每個八位元組之後，得到的'crc_value'的補碼以 LSB 優先順序排列並且本身[COBS]編碼。生成的編碼 CRC-32K 欄位的長度始終為五個八位元組。

收到訊號框後，'crc_value'被初始化為所有訊號框。編碼資料欄位的八位元組由解碼前的 calc_crc32K () 函數累加。然後對編碼的 CRC-32K 欄位進行解碼，並通過 calc_crc32K () 函數累積得到的四個八位元組。如果結果是預期的殘留值 'CRC32K_RESIDUE'，則正確接收訊號框。

下面顯示的示例 CRC-32K 功能用於說明。[BACnet]附件 G.3 中提供了使用 and 測試向量的完整示例。

<CODE BEGINS>

```
#include <stdint.h>

/* See BACnet Addendum 135-2012an, section G.3.2 */
#define CRC32K_INITIAL_VALUE (0xFFFFFFFF)
#define CRC32K_RESIDUE (0x0843323B)
/* CRC-32K polynomial, 1 + x**1 + ... + x**30 (+ x**32) */
#define CRC32K_POLY (0xEB31D82E)

/*
 * Accumulate 'data_value' into the CRC in 'crc_value'.
 * Return updated CRC.
 */
/* Note: crc_value must be set to CRC32K_INITIAL_VALUE
 * before initial call.
 */
uint32_t
calc_crc32K (uint8_t data_value, uint32_t crc_value)
{
    int b;

    for (b = 0; b < 8; b++) {
        if ((data_value & 1) ^ (crc_value & 1)) {
            crc_value >>= 1;
```

```

    crc_value ^= CRC32K_POLY;
  } else {
    crc_value >>= 1;
  }
  data_value >>= 1;
}
return crc_value;
}

```

<CODE ENDS>

附錄 D. 示例 6 LoBAC 訊號框解碼

本附錄僅供參考，不屬於本標準的一部分。

BACnet MS/TP, Src (2), Dst (1), IPv6 Encapsulation

Preamble 55: 0x55

Preamble FF: 0xff

Frame Type: IPv6 Encapsulation (34)

Destination Address: 1

Source Address: 2

Length: 537

Header CRC: 0x1c [correct]

Extended Data CRC: 0x9e7259e2 [correct]

6LoWPAN

IPHC Header

011. = Pattern: IP header compression (0x03)

...1 1... = Traffic class and flow label:

Version, traffic class, and flow label
compressed (0x0003)

.... .0.. = Next header: Inline

.... ..00 = Hop limit: Inline (0x0000)

.... 1... = Context identifier extension: True

....1.. = Source address compression: Stateful

....01 = Source address mode: 64-bits inline (0x0001)

.... 0... = Multicast address compression: False

....1.. = Destination address compression: Stateful

....10 = Destination address mode: 16-bits inline (0x0002)

0000 = Source context identifier: 0x00

.... 0000 = Destination context identifier: 0x00 [Source context: aaaa:: (aaaa::)]

[Destination context: aaaa:: (aaaa::)]

Next header: ICMPv6 (0x3a)

Hop limit: 63

Source: aaaa::1 (aaaa::1)

Destination: aaaa::ff:fe00:1 (aaaa::ff:fe00:1)

Internet Protocol Version 6, Src: aaaa::1 (aaaa::1),

Dst: aaaa::ff:fe00:1 (aaaa::ff:fe00:1)

0110 = Version: 6

.... 0000 0000 = Traffic class:
0x00000000

.... 0000 00.. = Differentiated

Services Field:

Default (0x00000000)

....0. = ECN-Capable Transport
(ECT): Not set

....0 = ECN-CE: Not set
.... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
Payload length: 518
Next header: ICMPv6 (58)
Hop limit: 63
Source: aaaa::1 (aaaa::1)
Destination: aaaa::ff:fe00:1 (aaaa::ff:fe00:1)
Internet Control Message Protocol v6
Type: Echo (ping) request (128)
Code: 0
Checksum: 0x783f [correct]
Identifier: 0x2ee5
Sequence: 2
[Response In: 5165]
Data (510 bytes)
Data: e4dbe8553ba0040008090a0b0c0d0e0f1011121314151617...
[Length: 510]

Frame (547 bytes):

55 ff 22 01 02 02 19 1c 56 2d 83 56 6f 6a 54 54	U.".....V-.VojTT
54 54 54 54 57 54 56 54 d5 50 2d 6a 7b b0 5c 57	TTTTTWTVT.P-j{.\W
b1 8e bd 00 6e f5 51 ac 5d 5c 5f 5e 59 58 5b 5a	n.Q.] _^YX[Z
45 44 47 46 41 40 43 42 4d 4c 4f 4e 49 48 4b 4a	EDGFA@CBMLONIHKJ
75 74 77 76 71 70 73 72 7d 7c 7f 7e 79 78 7b 7a	utwvqpsr} .~yx{z
65 64 67 66 61 60 63 62 6d 6c 6f 6e 69 68 6b 6a	edgfa`cbmlonihkj
15 14 17 16 11 10 13 12 1d 1c 1f 1e 19 18 1b 1a	
05 04 07 06 01 00 03 02 0d 0c 0f 0e 09 08 0b 0a	
35 34 37 36 31 30 33 32 3d 3c 3f 3e 39 38 3b 3a	54761032=<?>98;:
25 24 27 26 21 20 23 22 2d 2c 2f 2e 29 28 2b 2a	%\$' &! #"-,/.) (+*
d5 d4 d7 d6 d1 d0 d3 d2 dd dc df de d9 d8 db da	
c5 c4 c7 c6 c1 c0 c3 c2 cd cc cf ce c9 c8 cb ca	
f5 f4 f7 f6 f1 f0 f3 f2 fd fc ff fe f9 f8 fb fa	
e5 e4 e7 e6 e1 e0 e3 e2 ed ec ef ee e9 e8 eb ea	
95 94 97 96 91 90 93 92 9d 9c 9f 9e 99 98 9b 9a	
85 84 87 86 81 80 83 82 8d 8c 8f 8e 89 88 8b 8a	
b5 b4 b7 b6 b1 b0 b3 b2 bd bc bf be b9 b8 bb ba	
a5 a4 a7 a6 a1 a0 a3 a2 ad ac af ae a9 a8 ab aa	
ab 54 57 56 51 50 53 52 5d 5c 5f 5e 59 58 5b 5a	.TWVQPSR] _^YX[Z
45 44 47 46 41 40 43 42 4d 4c 4f 4e 49 48 4b 4a	EDGFA@CBMLONIHKJ
75 74 77 76 71 70 73 72 7d 7c 7f 7e 79 78 7b 7a	utwvqpsr} .~yx{z
65 64 67 66 61 60 63 62 6d 6c 6f 6e 69 68 6b 6a	edgfa`cbmlonihkj
15 14 17 16 11 10 13 12 1d 1c 1f 1e 19 18 1b 1a	
05 04 07 06 01 00 03 02 0d 0c 0f 0e 09 08 0b 0a	
35 34 37 36 31 30 33 32 3d 3c 3f 3e 39 38 3b 3a	54761032=<?>98;:
25 24 27 26 21 20 23 22 2d 2c 2f 2e 29 28 2b 2a	%\$' &! #"-,/.) (+*
d5 d4 d7 d6 d1 d0 d3 d2 dd dc df de d9 d8 db da	
c5 c4 c7 c6 c1 c0 c3 c2 cd cc cf ce c9 c8 cb ca	
f5 f4 f7 f6 f1 f0 f3 f2 fd fc ff fe f9 f8 fb fa	
e5 e4 e7 e6 e1 e0 e3 e2 ed ec ef ee e9 e8 eb ea	
95 94 97 96 91 90 93 92 9d 9c 9f 9e 99 98 9b 9a	
85 84 87 86 81 80 83 82 8d 8c 8f 8e 89 88 8b 8a	
b5 b4 b7 b6 b1 b0 b3 b2 bd bc bf be b9 b8 bb ba	
a5 a4 a7 a6 a1 a0 a3 a2 ad ac af ae a9 a8 50 cb	P.
27 0c b7	'..

Decoded Data and CRC32K (537 bytes):

78 d6 00 3a 3f 00 00 00 00 00 00 01 00 01 80	x...:?.
00 78 3f 2e e5 00 02 e4 db e8 55 3b a0 04 00 08	.x?...U;....
09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18	
19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28	 !"#\$%&' (
29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38	) *+, -./012345678
39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48	9:;<=>?@ABCDEFGH
49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58	IJKLMNOPQRSTUVWXYZ
59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68	YZ[\]^_`abcdefgh
69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78	ijklmnopqrstuvwxyz
79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 86 87 88	yz{ }~.....
89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 96 97 98	
99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 a6 a7 a8	
a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 b6 b7 b8	
b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 c6 c7 c8	
c9 ca cb cc cd ce cf d0 d1 d2 d3 d4 d5 d6 d7 d8	
d9 da db dc dd de df e0 e1 e2 e3 e4 e5 e6 e7 e8	
e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5 f6 f7 f8	
f9 fa fb fc fd fe ff 00 01 02 03 04 05 06 07 08	
09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18	
19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28	 !"#\$%&' (
29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38	) *+, -./012345678
39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48	9:;<=>?@ABCDEFGH
49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58	IJKLMNOPQRSTUVWXYZ
59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68	YZ[\]^_`abcdefgh
69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78	ijklmnopqrstuvwxyz
79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 86 87 88	yz{ }~.....
89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 96 97 98	
99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 a6 a7 a8	
a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 b6 b7 b8	
b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 c6 c7 c8	
c9 ca cb cc cd ce cf d0 d1 d2 d3 d4 d5 d6 d7 d8	
d9 da db dc dd de df e0 e1 e2 e3 e4 e5 e6 e7 e8	
e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5 f6 f7 f8	
f9 fa fb fc fd fe ff 9e 72 59 e2	rY.

Decompressed 6LoWPAN IPHC (558 bytes):																
60	00	00	00	02	06	3a	3f	aa	aa	00	00	00	00	00	00	\.....:?......
00	00	00	00	00	00	00	01	aa	aa	00	00	00	00	00	00	
00	00	00	ff	fe	00	00	01	80	00	78	3f	2e	e5	00	02	x?....
e4	db	e8	55	3b	a0	04	00	08	09	0a	0b	0c	0d	0e	0f	...U;.....
10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	
20	21	22	23	24	25	26	27	28	29	2a	2b	2c	2d	2e	2f	!\"#\$%&'()*+,-./
30	31	32	33	34	35	36	37	38	39	3a	3b	3c	3d	3e	3f	0123456789;:<=>?
40	41	42	43	44	45	46	47	48	49	4a	4b	4c	4d	4e	4f	@ABCDEFGHIJKLMNO
50	51	52	53	54	55	56	57	58	59	5a	5b	5c	5d	5e	5f	PQRSTUVWXYZ[\]^_
60	61	62	63	64	65	66	67	68	69	6a	6b	6c	6d	6e	6f	`abcdefghijklmnopqrstuvwxyz
70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f	pqrstuvwxyz{ }~.
80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f	
90	91	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f	
a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af	
b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	ba	bb	bc	bd	be	bf	
c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf	
d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	db	dc	dd	de	df	
e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	ea	eb	ec	ed	ee	ef	
f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	fa	fb	fc	fd	fe	ff	
00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	
20	21	22	23	24	25	26	27	28	29	2a	2b	2c	2d	2e	2f	!\"#\$%&'()*+,-./
30	31	32	33	34	35	36	37	38	39	3a	3b	3c	3d	3e	3f	0123456789;:<=>?
40	41	42	43	44	45	46	47	48	49	4a	4b	4c	4d	4e	4f	@ABCDEFGHIJKLMNO
50	51	52	53	54	55	56	57	58	59	5a	5b	5c	5d	5e	5f	PQRSTUVWXYZ[\]^_
60	61	62	63	64	65	66	67	68	69	6a	6b	6c	6d	6e	6f	`abcdefghijklmnopqrstuvwxyz
70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f	pqrstuvwxyz{ }~.
80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f	
90	91	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f	
a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af	
b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	ba	bb	bc	bd	be	bf	
c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf	
d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	db	dc	dd	de	df	
e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	ea	eb	ec	ed	ee	ef	
f0	f1	f2	f3													

我們感謝[RFC4944]的作者和 IETF 6LoWPAN 工作組的成員；這份文件大量借鑒了他們的工作。Ralph Droms 和 Brian Haberman 從一開始就提供了不可或缺的指導和支援。Peter van der Stok，James Woodyatt，Carsten Bormann 和 Dale Worley 提供了詳細的評論。Stuart Cheshire 發明瞭非常聰明的 COBS 編碼。邁克爾·奧斯本（Michael Osborne）批判性地觀察到，分別對資料和 CRC32K 欄位進行編碼將允許在運行中計算 CRC。Alexandru Petrescu，Brian Frank，Geoff Mulligan 和 Don Sturek 提供了寶貴的意見。

作者資訊

Kerry Lynn (editor)
Verizon Labs
50 Sylvan Rd
Waltham, MA 02451
United States of America
Phone: +1 781 296 9722
Email: kerlyn@ieee.org

Jerry Martocci
Johnson Controls, Inc.
507 E. Michigan St
Milwaukee, WI 53202
United States of America
Email: jpmartocci@sbcglobal.net

Carl Neilson
Delta Controls, Inc.
17850 56th Ave
Surrey, BC V3S 1C7
Canada
Phone: +1 604 575 5913
Email: cneilson@deltaccontrols.com

Stuart Donaldson
Honeywell Automation & Control Solutions
6670 185th Ave NE
Redmond, WA 98052
United States of America
Email: stuart.donaldson@honeywell.com