第二章 即時傳送架構下News的防匿方法

2.1 News的運作方式

現今存在於TCP/IP網路上的News系統，不外乎CNEWS與INN兩種。其傳輸方式皆遵循NNTP(Network News Transfer Protocol)[1]，而傳輸文件之格式亦以RFC1036(Standard for interchange of USENET messages)爲依歸。下圖即爲在NNTP架構下News的運作方式：

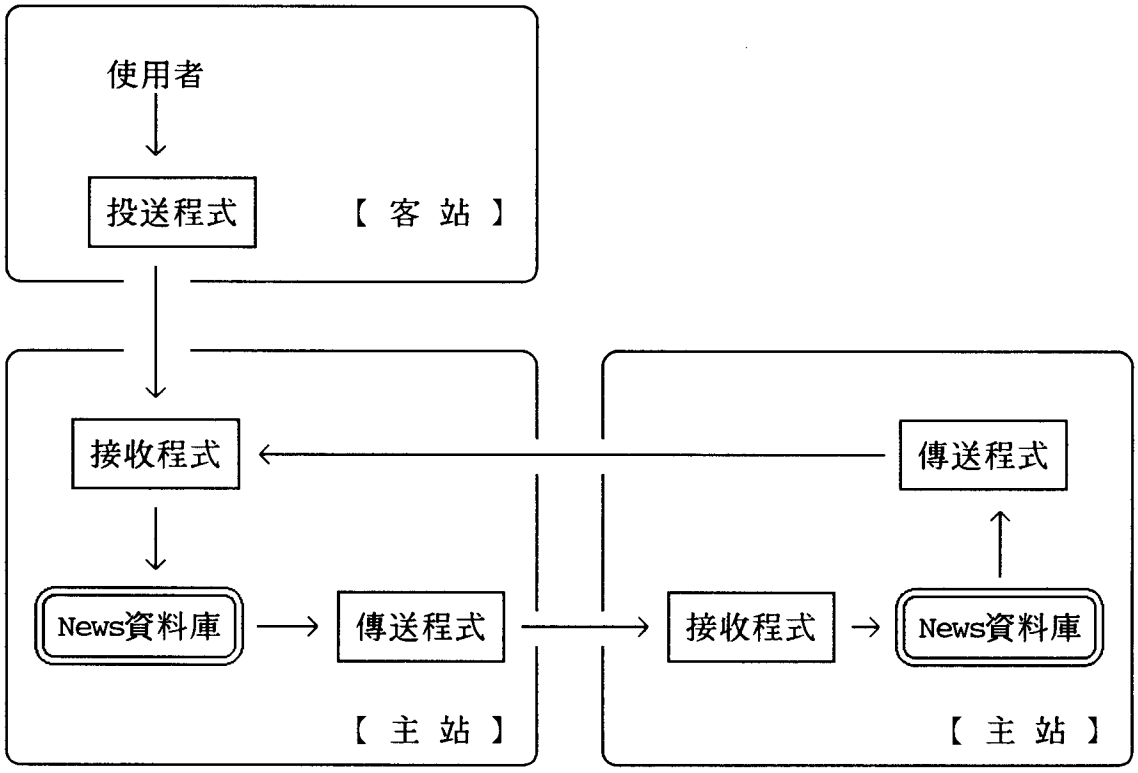


圖2.1 NNTP架構下News的運作方式

投送程式：如inews、tin等。

接收程式：如nntpd、in.nnrpd等。

傳送程式：如nntpxmit、nntplink等。

從上圖我們可以很清楚地看出News的防匿關鍵：在客站爲投送程式；在主站爲接收程式。

---

[1]詳見RFC977, Network News Transfer Protocol。

2.2 在News客站上的防匿方法

一篇使用者所寫的文稿從編寫完成到被安置在News主站前，News系統會先作如下的處理：

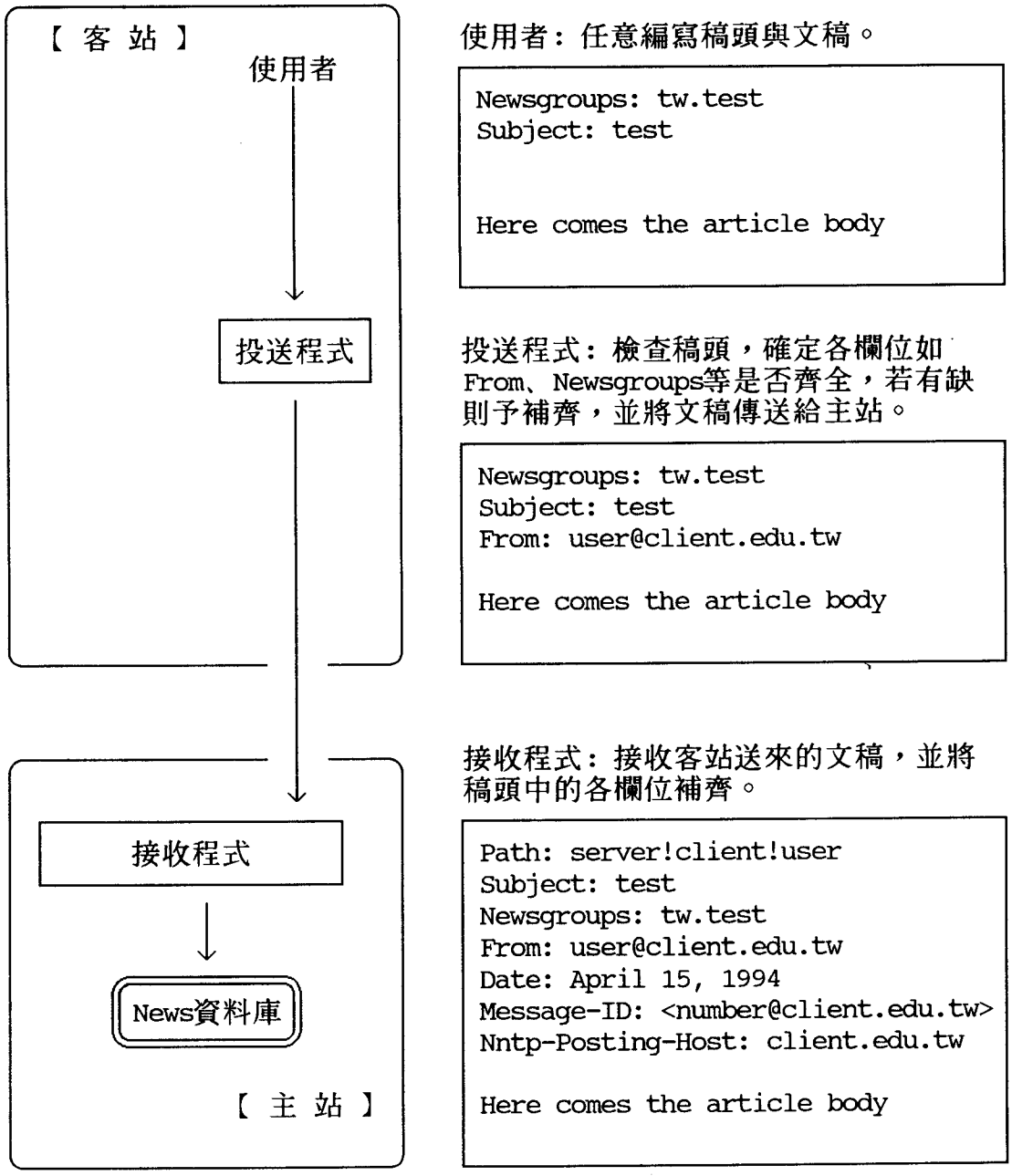
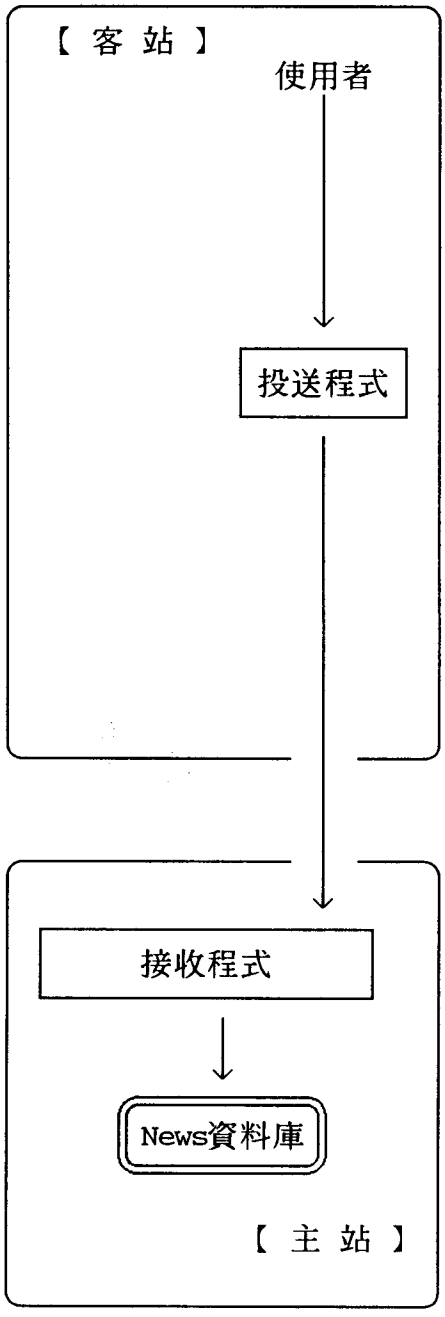


圖2.2 一般News主、客程式處理News的過程

從上圖我們可以看出一個給匿名者有機可乘的漏洞：由於投送程式完全接受使用者所編寫的文稿(包括稿頭中的From欄位)，所以From欄位的眞實性就非常值得懷疑。因此，我們必須對投送程式略加修正，限

定文稿中的From欄位完全由投送程式來填寫。亦即在投送程式收到使用者所寫的文稿後，要將稿頭中From欄位剔除。而文稿在傳送給主站前，投送程式再爲其補上正確的From欄位。下圖說明了在News客站上的防制程序：



使用者：任意編寫稿頭與文稿。

```
Newsgroups: tw.test
Subject: test
From: fake@client.edu.tw

Here comes the article body
```

投送程式：檢查稿頭，並將稿頭中的From剔除。確定各欄位如From、Newsgroups等是否齊全，若有缺則予補齊，並將文稿傳送給主站。

```
Newsgroups: tw.test
Subject: test
From: real@client.edu.tw

Here comes the article body
```

接收程式：接收客站送來的文稿，並將稿頭中的各欄位補齊。

```
Path: server!client!real
Subject: test
Newsgroups: tw.test
From: real@client.edu.tw
Date: April 15, 1994
Message-ID: <number@client.edu.tw>
Nntp-Posting-Host: client.edu.tw

Here comes the article body
```

圖2.3 具防匿功能的客端程式處理News的過程

2.3 在News主站上的防匿方法

上一節所述的簡易防制方法，只能應用在我們管理範圍內的客站。因爲對那些不在我們管理範圍內的客站而言，我們無法修正其投送程式。然而即使是在我們管理所及的客站上，我們仍然無法用修正後的投送程式來防止匿名文件的發生。因爲匿名者可以利用他自己的投送程式來投送匿名文章，根本不需理會我們所修正的投送程式。因此，強化主站上接收程式的功能，就變得格外地重要。

由於驗證協定(Identification Protocol)在TCP/IP網路上已逐漸被各種作業系統採用，而且很多伺服程式(server programs)已經具有和遠端驗證協定程式溝通的能力[1]。因此，我們將利用驗證程式來加強防匿措施——當文章被投送時，主站上的接收程式將向客站上的驗證程式查詢投送者的身份，並將其記錄於文章的稿頭中。其運作方式如下圖：

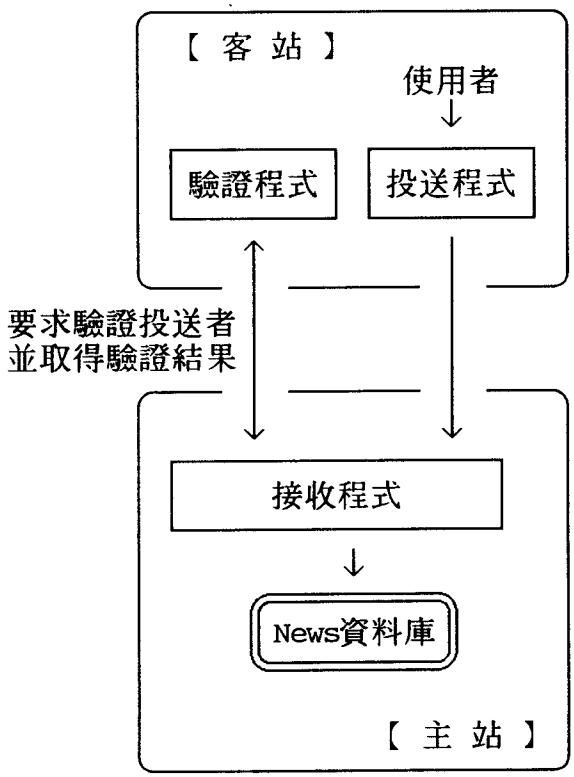
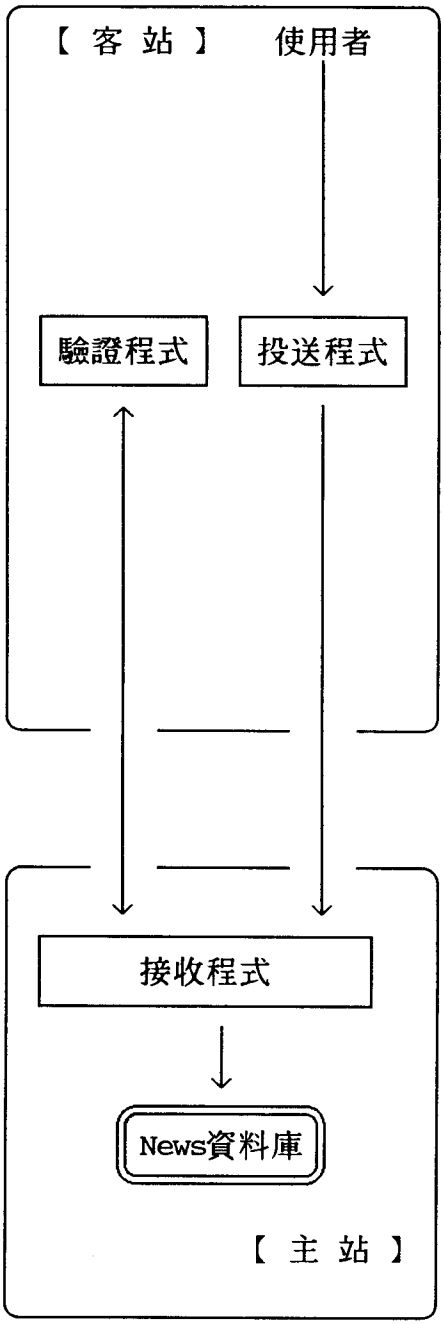


圖2.4 NNTP與驗證協定的運作模式

[1]例如tcpd— the tcp wrapper。

其運作步驟如下圖：



使用者：任意編寫稿頭與文稿。

```
Newsgroups: tw.test
Subject: test
From: fake@client.edu.tw

Here comes the article body
```

投送程式：檢查稿頭，並將稿頭中的From剔除。確定各欄位如From、Newsgroups等是否齊全，若有缺則予補齊，並將文稿傳送給主站。

```
Newsgroups: tw.test
Subject: test
From: real@client.edu.tw

Here comes the article body
```

接收程式：接收客站送來的文稿，並將稿頭中的各欄位補齊。向客站上的驗證程式查詢投送者的資料，且將查詢結果寫入Nntp-Posting-Host欄位中。

```
Path: server!client!real
Subject: test
Newsgroups: tw.test
From: real@client.edu.tw
Date: April 15, 1994
Message-ID: <number@client.edu.tw>
Nntp-Posting-Host: real@client.edu.tw

Here comes the article body
```

圖2.5 具防匿功能的主、客端程式處理News的過程

我們曾經評估過將投送者資料植入標準欄位From或Path的可行性，但由於From欄位的資料關係到客站隱域名稱(hidden domain)的設定與使用者別名(alias name)的使用，而Path欄位更牽涉到News主站間資料交換的效率，所以我們才選定Nntp-Posting-Host作爲填入投送者

資料的欄位。雖然Nntp-Posting-Host並非RFC1036所定義的標準欄位[1]，但是此欄位已經廣泛地被採用於News系統中。而且其原先所含的基本資料爲客站的地址，因此將投送者的資料加於此欄位是非常合適的。

對於沒有驗證程式的客站而言，本節所提出的防匿方法顯然無法奏效，但是我們可以用管制投送的方式來加強。例如我們可以預先在主站上設定某些客站的投送是可以接受的[2]，但要先確定這些客站都有驗證程式。或者我們不預先在主站上做任何限制投送的設定，但是當接收程式無法從客站取得投送者的資料時[3]，該篇文稿將不被接受。

雖然圖2.5所示的防匿措施已經能夠避免絕大部份匿名電子文件的發生，但是我們仍要注意某些讓匿名者有機可乘的漏洞：

- 接收程式處理稿頭的方式

  有些接收程式處理稿頭的方式是"照單全收，遇缺則補。"[4]在此情況下，匿名者只要投送一篇含有Nntp-Posting-Host欄位的文稿給主站，就能夠輕易地得逞。此外，Path、Date、Message-ID等欄位所含的資料對防匿工程來說是非常重要的，因爲我們可以從這些資料推算出匿名的時間與地點。因此，接收程式在處理稿頭時一定要對這些欄位嚴加管制。

- 接收程式對上游餵送源(newsfeed)的管制

  到目前爲止，我們一直都是針對投送(posting)模式中的各種缺點而提出各種防匿措施。對於熟悉NNTP協定的匿名者

---

[1]RFC1036定義的標準欄位只有From、Date、Newsgroups、Subject、Message-ID、Path。

[2]例如INN中的設定檔nnrp.access，NNTP中的設定檔nntp_access。

[3]發生這種情況的原因，主要是因爲客站沒有驗證程式；但也可能是因爲網路太過壅塞，投送者的資料無法在反應時間(response time)內傳回主站。當我們在使用驗證程式時，要特別注意第二種情況的發生。

[4]nntpd即是以此種方式處理稿頭，in.nnrpd則否。

而言，NNTP中的傳送(transmitting)模式提供了另一條可匿名的管道。如果我們在傳送模式下也用驗證程式來把關，那麼驗證程式將會投下大量的虛功去驗證彼端的傳送者[1]。幸好某些News系統可以指定上游餵送源[2]。如此一來，非餵送源上的使用者將無法再以傳送模式來匿名。

2.4 其它的防匿方法

根據前兩節的分析，我們知道任何嚴密的防匿措施都必須由主站與客站相互配合才能竟其功。準此要領，我們提出另外兩種方法：

- 秘密協定法

  我們可以讓主、客程式在NNTP的架構下，增加一個特別的協定，如HELO。若是主站在接受客站的投送前，沒有收到來自客站的HELO訊息，則客站所投送的文章將不會被主站接受。如此一來，我們才能保證主站所接受的文稿投送，都已經被具有防匿功能的客站投送程式處理過。

- 秘密欄位法

  客站在投送文稿時，必須在稿頭中加一個虛擬欄位如"Null:"。當主站收到含有此特殊欄位的文稿時，會將文稿收存，並將"Null:"剔除。反之，主站將不接受任何不含"Null:"欄位的文稿投送。如此我們才能保證主站所接受的文稿投送，都已經被具有防匿功能的客站投送程式處理過。

---

[1]亦即彼端的News系統管理者，如news。

[2]例如INN可以在hosts.nntp中指定上游的餵送源。