6J－06

複数 NIC とスイッチングハブを用いた広帯域通信機構の LINUX への実装*

東京電機大学 理工学部 情報システム工学科†
林 達馬　梅島 慎吾　桧垣 博章‡

1　背景と目的

近年、コンピュータネットワーク技術の発達によって、ネットワークに接続されたコンピュータ間で大量のデータを短時間で伝送することが必要となるネットワークアプリケーションへの要求が高まっている。特に、テキスト、音声、静止画像、動画像を含むマルチメディアデータを配送するアプリケーションとして、LAN をベースとした CSCW や、コラボレーションシステム [3] が研究開発されている。LAN の構築においては、イーサネット技術が広く利用されている。LAN には複数のコンピュータが接続されているため、CSMA/CD における競合と衝突の発生により、実効帯域幅はアプリケーションに対して必ずしも十分とは言えない。広帯域通信を実現するひとつの方法として、複数の通信路を束ねて利用するものがある。リピータハブを用いたイーサネットでは、各コンピュータに複数の NIC を装着しても、これらから送出されたパケットの伝送が競合、衝突してしまう。しかし、スイッチングハブを用いたイーサネットでは、各 NIC から送出されたパケットが独立に配送されることから競合や衝突が発生せず、広帯域な通信を実現することができる [1]。本論文では、これを実現するために必要となる拡張 ARP [4] およびこれを利用して広帯域通信を実現する機構の LINUX への実装について述べる。

2　広帯域通信の実現

これまでにも、限られた帯域幅を持つ通信メディアを複数用いて、送信元コンピュータと送信先コンピュータとの間で広帯域通信を実現する方法は広く用いられている。例えば、TCP/IP インターネットにおいては、バックボーンネットワークのルータ間の接続として複数の通信線を用い、これらのルータ間でパケット群を分割して配送する。また、ISDN 回線で PPP を用いて TCP/IP の通信を行なう場合、2B+D のうちの 2 つの B チャネル、あるいは 23B のうちのいくつかのチャネルを束ねることによる広帯域化が用いられている。しかし、これらの方法では固定の相手との通信を対象としている。LAN において複数の通信路を用いた広帯域化を行なうためには、各コンピュータに複数のイーサネットの NIC を装着し、通信要求に従って時々刻々変化する様々な相手との間に複数の通信路を用意しなければならない。これを目的としたものとして、LINUX の bonding device がある。ここでは、送信データは疑似デバイスに渡される。この疑似デバイスの機能によって送信データが複数のデバイスドライバに分割される。bonding device では、送信元コンピュータにおいて送信負荷を複数の NIC に分散することは行なっているものの、送信されたイーサネットフレームは送信先コンピュータに装着されたすべての NIC によって受信されることから、実質的に帯域幅を広げることができず、スイッチングハブなどに特別な機能を導入する必要性が指摘されている [5]。本論文では、複数の NIC を装着したコンピュータをスイッチングハブに図 1 に示すように接続することにより、これらのコンピュータ間での広帯域通信を実現する。このとき、以下の要求条件を満たすものとしなければならない。

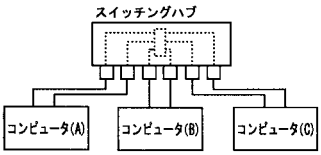

図 1: 複数枚 NIC を用いた広帯域通信機構

[要求条件]

(1) 提案する機構を導入しても、既存のアプリケーションへの変更を行なうことがなくそのまま利用できる。これを実現するために、同一のコンピュータに装着され、同一のネットワークに接続する複数の NIC(これらの MAC アドレスは異なる) には、同一の IP アドレスを与える。

(2) ネットワークに接続されているすべてのコンピュータに提案する機構が導入されていることを前提としない。提案機構が導入されているコンピュータと導入されていないコンピュータとが混在していても、TCP/IP による通信が正しく行なわれるものとする。

3　LINUX カーネルへの実装

2 章で述べた広帯域通信実現手法を LINUX カーネル (カーネルバージョン 2.2.17-0vl10) に実装する。実装にあたっては、以下の機構を実現する。

- IP 層からの送信要求を受けたデータリンク層において、送信すべき IP データグラムを複数の NIC のデバイスドライバに振り分ける制御機構 (スケジューラ)。なお、受信に関しては、IP が配送順序を保障しないプロトコルであることから、特別な機構を導入する必要がない。
- ひとつの IP アドレスを複数の MAC アドレスに対応付けることが可能な拡張 ARP プロトコル。
- ひとつの IP アドレスを複数の MAC アドレスに対応付けることが可能な ARP キャッシュ。

3.1　送信制御機構 (スケジューラ)

提案手法では、LINUX カーネルのデータリンク層で、送信に使用する NIC を選択するスケジューラ機能を実装する。LINUX カーネルでは、同一の LAN に接続する複数の NIC を装着すると、使用する NIC が明示的に指定されない場合、デフォルトの NIC を用いて送信が行なわれる。デフォルトの NIC とは、これらの NIC の中で各 NIC の情報が格納されているデバイス構造体のメンバ ifindex の値が最小である NIC である。デフォルト NIC を用いた送信要求に対してデータリンク層 dev.c では、どの NIC を用いて送信を行なうか決定され、送信に用いる NIC のデバイス構造体へのポインタを獲得し、以降はこの値を用いて送信処理が行なわれる。LINUX カーネルは、デバイス構造体のメンバ `hard_start_xmit` で指定されているフレーム送信関数を用いる。この関数

*Implementation of Extended ARP and High Performance LAN Conmunication in Linux Kernel.
†Tokyo Denki University
‡Tatuma Hayashi, Shingo Umeshima and Hiroaki Higaki

は、デバイスドライバの一部である。このように、送信に用いるデバイス構造体へのポインタの獲得手続きのみが変更されており、各デバイスドライバには変更がなされていないことから、使用する NIC のベンダに依存せず、フレームごとに異なる NIC を利用して送信することができる。

3.2 拡張 ARP の実装

前節で述べた複数 NIC による送信機構に加えて、複数の NIC による受信を実現するためには、送信先の IP アドレスを複数の MAC アドレスに対応付ける拡張 ARP [4] の実装が必要である。そこで、arp.c、neighbour.c、neighbour.h に変更を行った。ARP リクエストおよび ARP リプライを送信する関数 `arp_send` に、以下の処理を追加した。

- 従来の ARP メッセージフォーマットに加え、論文 [4] で述べた拡張部分を持つ ARP メッセージを送信する。拡張部分には、送信元と送信先の NIC の数、送信元と送信先の MAC アドレスのフィールドが含まれる。

また、ARP リクエストおよび ARP リプライを受信する関数 `arp_rcv` には以下の処理を追加した。

- 現在の LINUX カーネルでは、ブロードキャストで送信された ARP リクエストを複数の NIC が受信し、各 NIC が ARP リプライを送信してしまう。そこで、デフォルト NIC のみが ARP リクエストの処理を行なう。デフォルト NIC 以外は、ARP リクエストを破棄する。
- ARP リクエストの拡張部分からも、MAC アドレスを取り出す。

3.3 拡張 ARP キャッシュの実現

ARP では、ARP リクエストと ARP リプライの送受信によって得られた IP アドレスと MAC アドレスとの対応関係を ARP テーブルというキャッシュに保存する。以降の問い合わせに対して、キャッシュのエントリから得られた情報を用いて応答することによって、ARP メッセージのブロードキャストトラフィックを削減することができる。LINUX では ARP テーブルの各エントリはひとつの IP アドレスに対応しており、その情報は neighbour 構造体に格納され、管理されている。従来の ARP では 1 エントリにひとつの MAC アドレスが対応付けられている。しかし、拡張 ARP では、複数の MAC アドレスを格納する必要がある。従来の ARP では、neighbour 構造体のメンバ ha に MAC アドレスそのものを格納していた。拡張 ARP では、可変数の MAC アドレスを格納するために、ha を複数の MAC アドレスが格納されている配列の先頭アドレスへのポインタとする (図 2)。さらに、NIC の数を格納する neighbour 構造体のメンバ `nic_counts` を追加する。ひとつの neighbour 構造体に複数の MAC アドレスを対応付けることによって neighbour 構造体のメンバ `primary_key` に格納される IP アドレスをキーとして neighbour 構造体を探す関数 `__neigh_lookup` を使い、3.1 で述べたスケジュールの際、イーサネットフレームの送信先 MAC アドレスを送信元でスケジュールすることができる。

4 性能評価

提案手法の性能を評価するために、ネットワーク帯域幅を netperf [2] を用いて測定した。netperf は送信元ホストと送信先ホストで起動したプロセス間で UDP データグラムを送受信することによって、ネットワークの帯域幅を測定する。測定には、Pentium Ⅲ 1GHz、256MByte メモリのパーソナルコンピュータを用いた。また、100Base-T の NIC として、3Com 社製 3c59x と planex 社製 FNW-9700T、FNW-9802T を用いた。比較のため 1000Base-T の NIC として TP83820GB-PCI64 を用いた測定を行なった。なお、スイッチングハブには、ギガビットスイッチである FXG-04TE を用いた。測定結果を表 1 に示す。

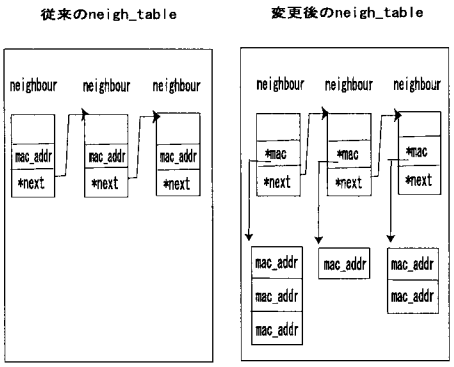

図 2: ARP テーブルの構造

表 1: 100Base-T を用いた帯域幅測定結果 (Mbps)

NIC 対	1	2	3
bonding device	93.29	96.61	98.37
デバイスドライバ変更 [1]	94.68	157.08	–
カーネル変更 (提案方式)	94.68	189.37	279.37

提案手法は、従来手法と比べて広い帯域幅を実現していることが分かる。1 対の 100Base-T の NIC によって、変更のないデバイスドライバと OS カーネルを用いて通信した場合の帯域幅は 93.14Mbps であったことから、提案手法は十分実用的なオーバヘッドしか要さないことが分かる。また、TP83820GB-PCI64 を用いた場合の帯域幅が 469.88Mbps であったことから、本手法の有効性が明らかとなった。

5 まとめ

複数 NIC を用いた広帯域通信機構を LINUX カーネルに実装した。今後は、スループットを向上させるスケジューリング方法を考案する。また、特定のネットワークアプリケーションを対象としたチューニングを行なう。

参考文献

[1] 出口, 桧垣, “複数 NIC とスイッチングハブを用いた広帯域通信機構の構築と評価,” 情報処理学会第 62 回全国大会, No. 1, pp. 25–26 (2001).

[2] Jones, R., “netperf,” http://www.cup.hp.com/netperf/NetperfPage.html.

[3] 木原, 藤井, 中村, 安斎, “ネットワーク共有空間での人間の動きによる描画と演奏,” 情報処理学会論文誌, Vol.40, No. 9, pp. 3501–3509 (1999).

[4] 中山, 林, 梅島, 桧垣 “複数 NIC とスイッチングハブを用いた広帯域通信のための拡張 ARP プロトコル,” 情報処理学会第 64 回全国大会, 6J–05 (2001).

[5] “road blancing,” http://webclub.kcom.ne.jp/ma/t-nagasa/Yotaro/butu_pc_h_lan_002a.html.