社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－ＭＢＬ－37 (16)
2006／5／19

メガノードマルチキャストシミュレーションに向けての送信実験報告

三角真† 中川晋一†,†† 我如古津世史†† 知念賢一†† 篠田陽一††

† 独立行政法人 情報通信研究機構
〒 184-8795 東京都小金井市貫井北町 4-2-1
†† 北陸先端科学技術大学院大学
〒 923-1292 石川県能美市旭台 1-1
E-mail: †{misumi,snakagaw}@nict.go.jp, ††{tganeko,k-chinen,shinoda}@jaist.ac.jp

あらまし インターネットを用いて大量のセンサ情報を複数地点で同時取得する通信方法として，IP マルチキャストが合理的である．しかし，IP マルチキャストを用いて 1 万を越えるデータを複数地点でルーティングし取得できたという報告は世界的になされていない．評価・実証にはシミュレーション環境が必要だが，実験に供する数万の送信ノード群，高性能のルーティング機構と受信・解析のシステム構築が必要となる．我々は，これらについて送信ノードの多重化 (1 台の実 PC で何台分の仮想ノードが実装可能か)，受信ノードの性能評価 (何万の送信ノードからの通信が受信可能か) の検討を行い，ルーチング機構として最も軽いレイヤ 2 スイッチングを用いて実験環境 StarBED を用いて合計 1 メガノード (100 万) の通信に成功した．実験途上で遭遇した様々な問題と解決法について報告する．

キーワード 性能評価，ベンチマーク，ユビキタスコンピューティング，大規模ネットワークテストベッド

Experiment Report of Mega-order Sensor Node Multicast Simulation

MISUMI MAKOTO†, NAKAGAWA SHIN-ICHI†,††, GANEKO TSUYOSHI††, CHINEN KENICHI††, and SHINODA YOICHI††

† National Institute of Information and Communications Technology
4-2-1, Nukui-Kitamachi Koganei, Tokyo 184-8795 Japan
†† Japan Advanced Institute of Science and Technology
1-1, Asahidai Nomi, Ishikawa 923-1292 Japan
E-mail: †{misumi,snakagaw}@nict.go.jp, ††{tganeko,k-chinen,shinoda}@jaist.ac.jp

Abstract IP multicast is the most efficient transport method for many to many data, especially in case of ten thousand sending and receiving pairs. We managed to succeed as the first case in the world for Mega-order send-receive pairs using L2 IP multicast, prosess-mutiplexing condition at 'StarBED.' In this report, some results and asessments from this experiment were investigated.

Key words Performance evaluation, Benchmark, Ubiquitous computing, Large-scale network test bed

1. はじめに

インターネットを用いたセンサーネットワークの通信手法として，IP マルチキャストが運用上最も合理的である．微小センサノードからの通信を想定し，複数の受信ノードに対する任意地点へのデータ配送においては，自由度の高い IP マルチキャストが有効である．[1], [4], [6] 従来の IP マルチキャスト通信に関する研究は，放送の様に単一の送信ホストから多数の受信ホストに対するものであり，検討されてきた内容は，主に中継ノード上でのパケットメモリコピーとマルチキャストツリー構造に関するものなどであり，数万を越えるような打ち上げが行なわれる場合のマルチキャストネットワークの挙動や伝送に関する要件は明らかにされていない．今回，IP マルチキャストのスケールに関する諸問題を明らかにするため，大規模ネットワーク実験設備である StarBED を用いてメガオーダーマルチキャストネットワークの実実験環境を構築した．100 台程度の実 PC を用いて 1 メガノードを作成し，毎秒合計 1 メガパケットのパケットの送出を目的として実験を行なう環境構築に成功した．構築したシミュレーション環境の送信ノードの仮想化技術，および実験途上で生じた問題点について報告する．

2. 目的と方針

メガオーダを越えるマルチキャストパケットを打ち上げる(sender が 100 万をこえる)経験は未知である．ルーチングに関する問題として，スイッチの MAC テーブルルックアップの負荷，スイッチングエンジンの負荷，その他，通常の利用では予測できない要因が障害となって．受信側へのパケット到着間隔の延長 (ジッタの増大) やパケットロスが生じることが考えられる．また，今回想定する IP マルチキャストは一種の flooding type の broadcast traffic であり，送信ノード側のインターフェイスに対しても負荷となる．さらに 1 個 1 個のパケット長が 100 バイト程度であるパケットが毎秒 1 メガ個到着する場合のデータ量は 100 メガバイト/秒であり，最速のハードディスク (最速で 10Mbytes/sec, 2006 年) を用いても記録を取ることは不可能である．従って，ルーチング，送信側インタフェイスに対する余剰トラヒックのフィルタリング，受信機構の負荷分散と記録方法の検討が必要になる．

2.1 通信モデル

本研究では，現在想定されている接続されたセンサからの信号を符号化，データパケットを生成，受信側の状態を確認せずにデータを伝送する様な，微小ノードを用いたデータ収集を目的としたセンサネットワークのトラヒックを IP マルチキャスト (今回は特にツリー構造を考える事の無いように L2,IPv4) を想定した．

2.2 仮想化

大規模な IP マルチキャストの通信を想定した場合，1 メガノードからのパケットが流れたとしても，それらのエミュレーションの規模 (送信数の増大) により，伝送ネットワークにかかる負荷も変化する．本研究の目的は実験環境の構築と評価であり，一つ一つの送信ノードの送信精度，受信ノードの解析性能のベンチマークを行い，送信・受信間隔とパケットロスを尺度として，ネットワークの問題点を探索する必要がある．

特に，実験環境として，実際に 100 万ノードを実ノード (PC) を用いて用意することは現実的ではなく，仮想ノードを実ノードと送信間隔を基準にした送信精度の低下が許容範囲に収まる事を条件として，可能な限り一つの実ノードから多くのトラヒックを送信させることを検討した．以下，実ノードから多数のノードに通信させる事を仮想化と呼ぶ．

多重化の階層として，MAC 層での多重化，IP 層での多重化，UDP 層での多重化，アプリケーション層での多重化が考えられる．

多重化を行う層が，下位であるほど実際のパケットの挙動に近づき，また，それに伴い，同数のパケットがネットワークに流れたとしてもネットワークの負荷が大きくなると考えられる．

今回は，最も単純に多量のパケットを送出するために，プロセス多重を用いて，UDP 層での多重化を行い，パケットの送出を行なった．

本実験の様な，定期的にパケットを送出する単純なモデルを仮想化する場合，複数のノードをシミュレートするプログラムを作成するよりも，実際のプログラムそのものを複数動作させる事が容易であると考えた．そのため，定期的にパケットを送出するプロセスを 1 台の PC で複数動させ，それぞれのプロセスを 1 つの仮想ノードと見なした．

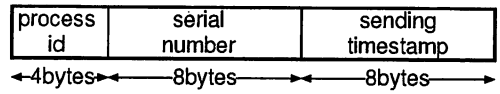

図 1 送信パケットのフォーマット

3. 実　　験

3.1 本実験の目的

メガオーダーの仮想ノードを実現し，それらの仮想ノードからパケットの送出・伝送の確認を目的とし，以下の実験を行った．

3.2 計測プログラム概要

3.2.1 send_udp

パケットを送信するためのプログラムで，図 1 に示すように，データ部分に，送信間隔を調査するための送信時のタイムスタンプ，多重化した際に送信プロセスを一意に判別するためのプロセス識別子，パケットの到着順序及び消失を検知するためのパケットのシリアル番号を持った UDP パケットを，指定した時間毎に送出する．今回の実験では，1 秒毎にパケットを送出する．

なお，多重化のために，複数の送信ノードを単一の PC でシミュレートするために，send_udp は，指定した数だけプロセスを実行する．

3.2.2 sara

当初，tcpdump や専用のプログラムを用いてパケットを HDD にダンプするという手法も試みたが，専用のプログラムの場合は，毎秒合計 5000 ノードからのパケットの受信で 0.1%程度，毎秒合計 30000 ノードからのパケットの受信で 1%程度のパケットドロップが認められた．そのため，パケットをキャプチャし，各ノードが送出したパケットの送信間隔と受信間隔の平均値及び標準偏差，ドロップしたパケット数を，ノード毎に算出し記憶し，終了時に各ノード毎の統計情報と，受信したパケット全てのノードの統計情報を出力する．Save and Recording Agent(sara) を作成した．

送信間隔及び受信間隔を求める際には，シリアル番号が非連続な場合，つまり，パケットドロップした場合は，図 2 に例を示すように，それらの間隔を平均値や標準偏差の導出に利用しない．そのため，パケットのドロップが送受信間隔の平均値と標準偏差に影響を及ぼすことは無い．

3.2.3 ami

A Multicast Introducer(ami) は，受信ノードが，マルチキャストグループに参加するためのプログラム．

3.3 実験ネットワークトポロジ

図 3 に，今回メガノード実験を行った際のネットワークを示す．

Group A に属する 1 台の PC を IGMP Querier とし，207

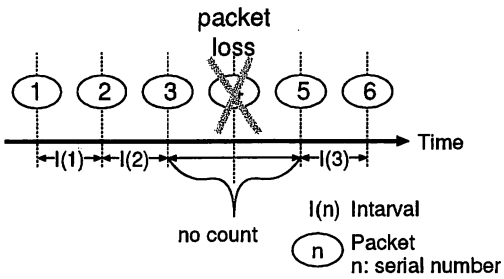

図 2　パケットドロップ時の送受信間隔の扱い

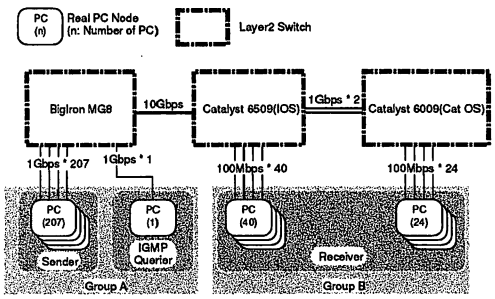

図 3　実験ネットワークトポロジ

表 1　実験に利用した PC のスペック

	GroupA (Sender and IGMP Querier)	GroupB (Receiver)
CPU	Pentium3 1GHz	Pentium3 1GHz
Memory	512MB	512MB
NetIF	1000Base-T	100Base-T
OS	Linux 2.6.13	Linux 2.6.13

台を送信のための PC として利用した．また，Group B に属する 64 台の PC を受信ノードとして利用した．

全ての PC は同一の Layer 2 ネットワーク上に存在する．なお，スイッチ間は Tagged VLAN を用いて通信する．

3.4 計測手法

実験の駆動は，宮地らの設計開発した SpringOS [2], [3] を用いて行った．実験流れを図 4 に示す．

PC をコントロールするために送信 PC と受信 PC で，“slave” を実行し，これらの PC をコントロールするためにコントロール PC で “mastasr” を用いる．なお，コントロール PC は，純粋に実験の駆動のみを行い，実験そのもののための通信は行わない．

送信 PC 及び受信 PC それぞれの実験用のシナリオを受け取り，ネットワークを設定する．続いて，受信 PC で，マルチキャストグループに参加するためのプログラム (ami) とパケットを受信し統計処理を行うプログラム (sara) を実行する．layer3 のネットワークが正常に設定されている事を確認するために，送信 PC から受信 PC に対して定期的に ping を打つ．ping の応答が有った時点でネットワークの設定が終了したと見なし，“master” に “Network was ready” のメッセージを送る．送信

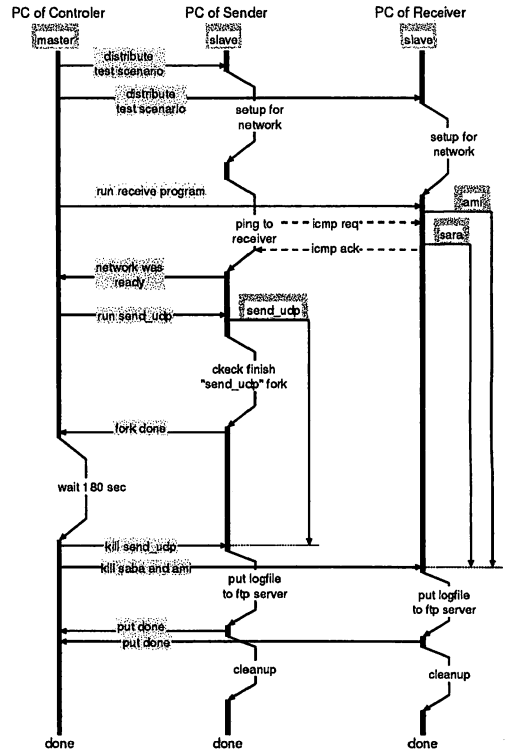

図 4　実験駆動の流れ

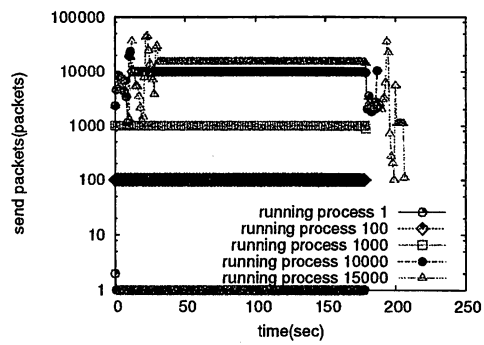

図 5　実行プロセス数と 1 秒当りの送出パケット数

PC でノードをエミュレートする send_udp を実行する．図 5 は，プロセスを実行し始めてからの 1 秒当りのパケット送出数を実行プロセス数毎に示したグラフであるが，これより多くのプロセスを実行した場合は，全てのプロセスの実行完了までに暫く時間が掛ることが分かる．そのため，全ての仮想ノードが出現してから観測を始めるために，指定した数のプロセスの起動を監視し，起動の完了を “master” に報告する．全ての送信 PC で，指定した数の仮想ノードを実行後，180 秒間の観測を行う．その後，送信 PC で send_udp を停止し，また，受信 PC で ami と sara を停止する．送信 PC と受信 PC は，ログ

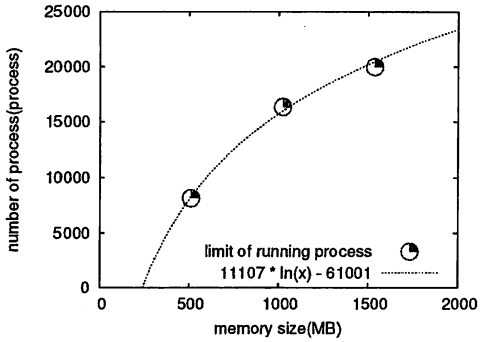

図 6　実装メモリの量と最大実行可能プロセス数の関係

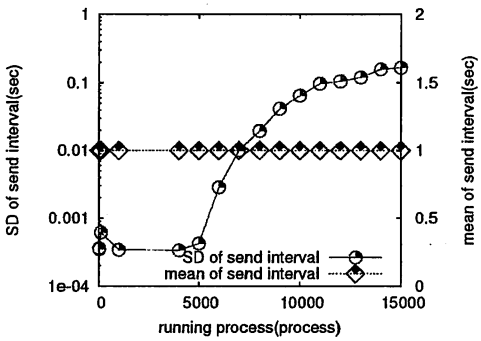

図 7　プロセスの多重度と送信間隔の平均値と標準偏差の関係

ファイルを ftp サーバに put し，その作業が終了すると終了した旨を "master" に報告，そして，実験のための設定を消去し実験シナリオを終了する．

4. 結果と考察

4.1 送信ノードの条件検討

表 1 に，今回実験に用いた送信ノードを示す．本実験では，PC1 台で 1 つのプロセスを実行した場合の送信間隔の平均値と標準偏差を基準とし，多重度を検討した．多重度は，複数の仮想送信ノードが実行される 1 台の PC と，受信ノードとなる 1 台の PC を用いて検討した．

PC で単一のプログラムを実行した際の送信間隔の標準偏差は，図 7 に示すように，数百マイクロ秒の精度であった．そこで，本実験では送信間隔の標準偏差が，想定する送信間隔の 1000 分の 1 である，1 マイクロ秒以下の場合を，正常な動作をしているとみなした．また，実験によって求めたメモリの量と最大実行プロセス数を表す図 6 を基に，表 1 の GroupA の PC で多重度の探索を行い，5000 の仮想送信ノード多重が可能であることを確認した．

4.2 受信ノードの条件検討

1 台の PC で実行した受信プログラムが，処理できる送信ノード数の上限を求めた．

当初，高性能な 10GbE 搭載の Dual Opteron PC で数十万

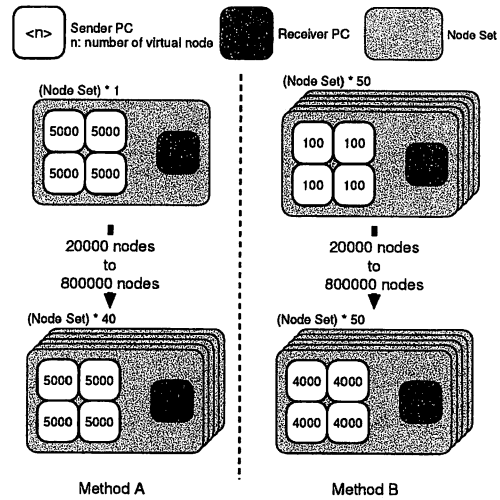

図 8　実験ノードの増加方式

ノードからのパケットを処理しようと考えていた．しかし，sara を用いて受信したところ毎秒合計 50000 ノードからのパケットを受信した時点で，sara が 0.5% のパケットを取りこぼしていることが分かった．また，調査したところ，sara の用いているパケットキャプチャライブラリである，libpcap で純粋にパケットのキャプチャを行うだけでも，0.5% のパケットの取りこぼしが発生することが分かった．当実験では，メガオーダーの送出されたパケットを十分に受信出来るノードを用意したいと考えたため，表 1 の GroupB を複数台用い，パケットを分散して受信する手法を用いた．

事前に，1 台の PC で多重可能なノード数が 5000 で有ると分かっていたため，送信仮想ノードを多重化した PC を 1 台づつ増やすことで，5000 刻みで送信ノードを増加し，単一の受信プログラムが処理可能なノード数を探索した．パケットのドロップ数を判断基準とし，恒常的にパケットをドロップしない台数を探索した．結果，表 1 の GroupB では 20000 台のノードまでのパケットを落とすことなく処理できる事を確認した．

4.3 送受信ノードセット

4.2 で述べたように，1 台の受信ノード (表 1 の GroupB) で，1 メガノードのパケットをモニタは困難であったため，20000 台の仮想ノードに対して 1 台の受信ノードを用いる．

1 メガオーダーを実現するための仮想ノードの増加の手順として，図 8 の様な手法が考えられる．図 8 に，両手法で 20000 台の仮想ノードと，800000 台の仮想ノードを実現する際の具体例を挙げる．

図 8 の Method A の場合は，1 送受信セットの，4 台の PC それぞれで 5000 ノードを仮想化し，仮想ノード数を変化させる場合は，このノードセットの数を増減する．実験を通して，パケットを送信する PC では常に 5000 ノードが多重化されており，また，受信ノードは常に毎秒合計 20000 パケットを処理する．そのため，実験で利用する仮想ノードの数を変えた計測

であっても，送受信ノードの精度はそれぞれの計測で同様である．そのため，送受信のための PC の精度に関しては，仮想ノード数に依存せず同様の条件で計測が可能である．

図 8 の Method B の場合は，1 セット (4 台の送信のための PC と 1 台の受信のための PC) を，あらかじめ 1 メガ仮想ノードを実現出来る 50 セット分を利用し，それぞれの送信のための PC での多重度を変化させることで，実験に用いる仮想ノードの数を変化させる．この手法は，送信ノードが少数の場合でも，実験に用いる全ての PC を利用するため，全体の仮想ノード数を変化させ計測した場合でも，実験を通して，実験ネットワーク全体対して均等に負荷を与えることが可能である．しかし，仮想ノードの数を変化するために，送信のための PC 上での多重度が変化すること，また，受信ノードで単位時間あたりに処理するパケット数が変化することから，計測値の微細な変化が，送信側，ネットワーク，或は受信側のいずれに起因するものかの特定が困難である．

当実験では，送受信の統計情報を取るために均質にパケットを増加したいと考えたため，MethodA を用いた．

4.4 L2 ルーチングの問題

以下に挙げるような問題の対策をし，同一 L2 ネットワークでメガオーダーのパケットを送出し伝送できる系を構築した．

4.4.1 フラッディングパケットの送信ノードへの影響

IGMP Snooping が有効でなく，マルチキャストパケットが同一 Layer2 ネットワークにフラッディングしてしまう場合，送出するパケットが少量であれば問題は発生しない．しかし，メガオーダーのノードからのパケットがフラッディングすると，それらのパケットを受信することによりネットワークインタフェースに負荷が掛り，パケットの送出に影響が及ぼされることが確認された．

これより，多量のマルチキャストパケットを用いて通信する場合，受信ノードの接続されたポート以外にパケットを送出しない，IGMP Snooping の様な機構が必要となる．

4.4.2 IGMP Querier の過負荷

今回，PC で mrouted を動作させ IGMP Querier とした．その際，毎秒メガオーダーのパケットが送出されると，それらのパケット全てがマルチキャストルータである IGMP Qureier に対して全てのマルチキャストパケットが到着する．そのため，ネットワークカードで多量の割り込みが発生し，OS が過負荷になり IGMP Query の送出が正常に行えない状態であった．今回の PC マルチキャストルータの役割は，マルチキャストパケットを転送することではなく，IGMP Query を定期的に出すことであった．そのため，ネットワークインタフェースのドライバオプションで単位時間あたりの割り込み回数を制限することで，OS の処理する受信パケットを制限し，IGMP Querier が過負荷になることを防いだ．

4.5 伝送品質の低下

1 メガノードからの毎秒合計 1 メガパケットの送出を行うことに成功した．

図 9 に受信間隔の平均値と標準偏差，図 10 に送信間隔の平均値と標準偏差を示す．これら 2 つの図は相似で有ることから，

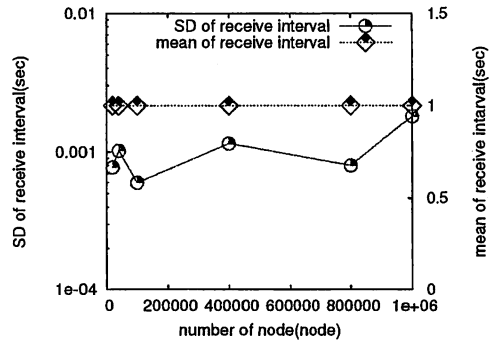

図 9 ノード数毎の受信間隔の平均値と標準偏差

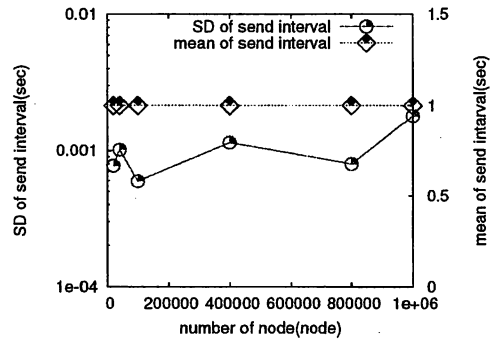

図 10 ノード数毎の送信間隔の平均値と標準偏差

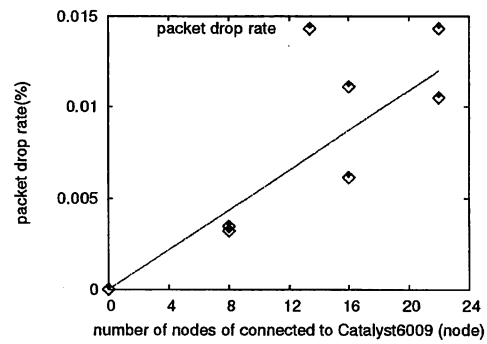

図 11 Catalyst6009 に接続された受信ノードの台数とパケットドロップ率の関係

ネットワーク上で遅延や揺らぎの影響を受けていないことが推測できる．

しかし，我々が [1] で述べたように，Catalyst6009 に接続された受信ノードを用いたことが原因であると思われる，パケットドロップが観測された．図 11 に，Catalyst6009 に接続されている受信ノード割合と，パケットロスの関係に関するグラフを示す．このグラフは，640000 仮想ノードからパケットを送出し，32 台の PC でパケットを受信し，Catalyst6509 と Catalyst6009 に接続される受信ノードの台数を変えて測定を

行ったものである．X 軸が，Catalyst6009 に接続された PC の台数で，Y 軸がパケットドロップ数を基に求めたパケットドロップ率である．つまり，ネットワーク上に流れているマルチキャストパケットの数は同様であるが，受信ノードの接続されたスイッチの割合が異なる．その結果，Catalyst6009 に接続された PC の台数と，パケットドロップの割合はほぼ比例していることが分かる．これより，Catalyst6009 内でのパケットドロップは，単位時間に通過するパケットの量に関わらず，一定の確率で起きていると推測される．

5. ま と め

送信ノードが 10^6 規模のマルチキャストパケットの送出・伝送を実現目的とし，これらを実現するための要件を探索するため，試行的に StarBED を用いて，プロセス多重で 1 メガノードをエミュレートし，合計毎秒 1 メガパケットをマルチキャストで送出し受信することに成功した．

メガノードを実現するために，単一の PC での多重度の検討を行った．また，単一の受信ノードでどれだけのパケットが受信可能であるかについても検討を行った．これらの結果から，送受信セットを作成し，セットを増減することで仮想ノード数を変更することで，一定の精度を保った送受信ノードを用意した．

複数の実 PC で実現される多数のノードの計測を行う際に，全てのノードからのデータを一定以上の時間するために，プロセスの起動のばらつきを考慮し，計測のスケジューリングを行った．

数万程度のマルチキャストパケットが同一の layer2 ネットワークにフラッディングすると，パケットの送出に影響が有ることが確認された．また，毎秒 1 メガのマルチキャストパケットがフラッディングすれば，これらのパケットを浴びた PC はパケットの送信さえ行えない状態になることが確認された．よって，メガオーダーのマルチキャストパケットを用いた通信を行う際には，IGMP Snooping の様なパケットをフラッディングしないための機構が必要であることが示唆された．

送受信間隔の平均値と標準偏差やパケットのドロップ数の統計情報から，ネットワークの障害や問題，ボトルネックリンクが推測できることが示唆された．

謝　　辞

本研究を行うにあたり，ご指導ご助言頂いた東京工業大学理工学研究科酒井善則博士，山岡克式博士，篠宮俊輔氏，情報通信研究機構大槻英樹博士，町沢朗彦氏，インテック北口善明氏，ならびに諸氏に深謝する．また，本研究は情報通信研究機構運営費交付金 (情報通信部門)，JGN2-A17003,「IP マルチキャストを用いた Simple Node Administration Protocol 実験」，NiCT 北陸 IT 研究開発支援センター，平成 17 年度厚生労働省がん研究助成金研究総合研究「がん情報ネットワークを利用した総合的がん対策支援の具体的方法に関する研究」若尾班等の支援，Cisco Systems Japan ならびに三井物産株式会社の技術協力を得て行った．関係各位に深謝する．パケット受信プログラム sara を作成するにあたり，参考にさせて頂いた ENMA [8] の制作者である中村豊氏に感謝します．

文　　献

[1] 三角真，中川晋一，我如古津世史，知念賢一，篠田陽一: "メガノード IP マルチキャストセンサネットに関する実験的検討", 信学技報，CQ2006016(2006-4)，pp.75-81，ISSN 0913-5685，2006

[2] Toshiyuki Miyachi, Ken-ichi Chinen and Yoichi Shinoda: "Automatic Configuration and Driving Internet Experiments On An Actual Node-base Testbed", Tridentcom2005, pp.274-282, Trento, ISBN 0-7695-2219-X, 2005

[3] 宮地 利幸，知念 賢一，篠田 陽一: 「SpringOS/VM: 大規模ネットワークテストベッドにおける仮想機械運用技術」，情処研報，2005-OS-99，Vol.2005，No.48，pp.105-112，恩納，情報処理学会，ISSN 0919-6072，2005

[4] 三角真，中川晋一，我如古津世史，知念賢一，出口眞人，篠田陽一: 「メガオーダノードセンサネットエミュレーションに関する実験的検討」，The Seventh Workshop on Internet Technology - WIT2005，2005.11.

[5] Takemoto, S., T. Yamamoto, A. Mukai, S. Otsuka and K. Fujimori: Crustal Strain Observation for Nine Years with a Laser Strainmeter in Kobe, Japan, Jounal of Geodynamics, Vol. 35/4-5, (2003) 483 - 498.

[6] 出口眞人，中川晋一，三角真，篠田陽一: マルチキャストを用いたモバイルコンピュータデバイス拡張の提案，IPSJ-SIG Technical Report 2004-MBL-31(21),pp157-162，2004

[7] 根日屋英之，植竹古都美: ユビキタス無線工学と微細 RFID，東京電機大学出版局，2004

[8] 中村 豊，知念 賢一，砂原 秀樹，山口 英: ENMA: パケットモニタによる WWW サーバの性能計測システムの設計と実装，電子情報通信学会 和文論文誌 D-I，p329-338,2000.3.