

Managed M2M システムの性能評価

峯村治実[†] 金子洋介[†]
安田晃久[†] 鶴川達也[†]

当社で開発した Managed M2M システム技術は、機器の遠隔監視・制御をインターネット経由で実現する技術であり、即時制御・安全性・キャリア非依存が特長である。本技術では、監視・制御対象の多数の機器と接続することになるため、最大接続数を評価した結果、数千台の機器を接続しても性能的に問題ないことが確認できた。また、データサイズと応答性能についても評価を行い、応答時間が送受信データサイズにほぼ比例するという結果が得られた。

Performance Evaluation of Managed M2M System

Harumi Minemura[†] Yosuke Kaneko[†]
Akihisa Yasuda[†] Tatsuya Tsurukawa[†]

Managed M2M system technology, that we have developed, is a technology to achieve remote monitoring and control of various devices via the Internet. The features of the technology are immediate access to the devices, safeness of the communication, and carrier independence of the protocol. Because the devices that are monitored or controlled are always connected in this technology, we evaluated the number of the simultaneous connections of the devices, and we confirmed that thousands of devices could be connected without any performance loss. We also evaluated the response performance of the technology, and we got the result that the response time was approximately proportional to the size of the data to be transmitted and received.

1. はじめに

近年、ネットワークインタフェースと TCP/IP プロトコルスタックを備えた組み込み機器が増加しており、ネットワーク経由でこれらの機器の遠隔監視・制御を行うことが可能となってきている。これにより、遠隔地に設置されたさまざまな機器の保守点検をネットワーク経由で行うことが可能となり、問題解決の迅速化・保守コストの低減が期待できる。

ネットワーク経由で機器の遠隔監視・制御を行うためには、監視・制御を行う監視センターと監視・制御対象の機器との間を広域網で接続する必要があるが、広域網として専用線や VPN を用いるとコストが高くなり、運用や設定が煩雑になるという課題がある。一方、広域網としてインターネットを用いれば、コストは安く、運用や設定も容易であるが、監視・制御対象機器のアドレスをインターネット上に公開する必要があり、外部からの攻撃に対して脆弱となるため、安全性の面で問題がある。

我々が開発した Managed M2M システム技術[1][2][3] (M2M: Machine-to-Machine) は、このような課題を解決する技術である。本技術では、監視センターと監視・制御対象機器との間を直接ネットワークで接続するのではなく、M2M サーバと呼ぶ中継サーバを経由して接続する。これにより、監視センターも監視・制御対象機器もアドレスを外部に公開する必要がなくなり、外部からの攻撃を受けにくくなるため、インターネット経由の安全な遠隔監視・制御を実現することができる。

Managed M2M システム技術では、監視・制御対象機器と M2M サーバとの間を常時接続しておく必要がある。このため、多数の機器の遠隔監視・制御を行う場合には、一般的な Web サーバ等と比較して M2M サーバの負荷が高くなることが予想される。また、本技術では、監視センターから送信したコマンドも、それに対して監視・制御対象機器が送信する応答データも、一度 M2M サーバのメモリ上に保持されるため、送受信データ量が大きくなると、応答時間が増大することが予想される。

本稿では、実用的に問題がないかの確認を目的として、Managed M2M システム技術で多数の機器を接続した場合の性能と、大容量のデータを送受信した場合の応答性能を測定した結果について報告する。

2. Managed M2M システム技術の概要

M2M (Machine-to-Machine) は、機器と機器がネットワークを経由して相互に通信を行う通信形態を表す用語である。Managed M2M システム技術では、M2M サーバで

[†] 三菱電機株式会社 情報技術総合研究所
Mitsubishi Electric Corporation, Information Technology R&D Center

この機器間の通信を中継することによって安全な通信を実現しており、機器間の通信を管理 (manage) しているため、我々は“Managed”と称している。

図1に Managed M2M システムの構成を示す。Managed M2M システムの構成要素としては、前述の M2M サーバの他に M2M クライアントと M2M エージェントがある。以下、これらの構成要素について説明する。

- M2M クライアント：監視・制御端末上で動作し、監視・制御アプリケーションから受け取った処理要求を M2M サーバ経由で M2M エージェントに送信する。また、M2M エージェントから M2M サーバ経由で受け取った処理結果を、監視・制御アプリケーションに返す。監視・制御端末はファイアウォールの内側に設置可能である。一般的には、複数の M2M クライアントを含む複数のサイトがファイアウォールを経由してインターネットに接続される。
- M2M エージェント：機器上または複数の機器を一括して監視・制御する監視・制御装置上で動作し、監視・制御アプリケーションから M2M サーバを経由して受け取った処理要求を機器固有のソフトウェアに渡して、要求された処理を実行させる。また、機器固有ソフトウェアの処理結果を M2M サーバを経由して M2M クライアントに送信する。各機器はファイアウォールの内側に設置可能である。M2M クライアントの場合と同様、複数の M2M エージェントを含む複数のサイトがファイアウォールを経由してインターネットに接続される。

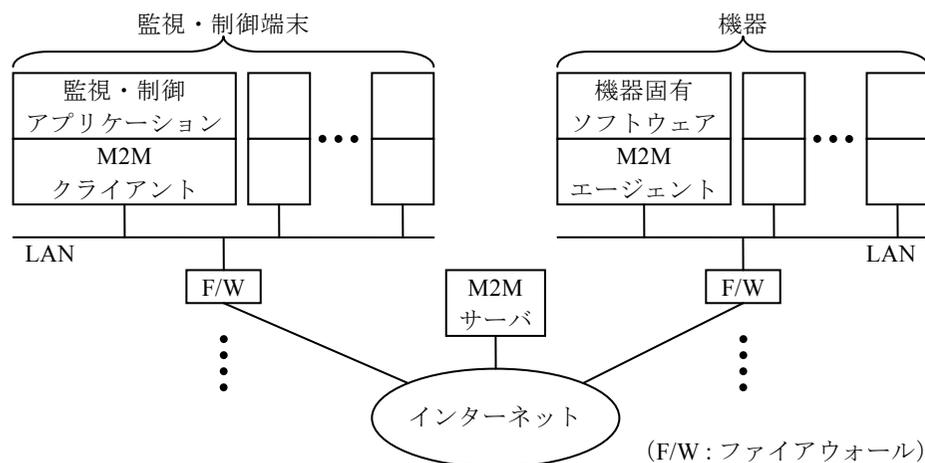


図1 Managed M2M システムの構成

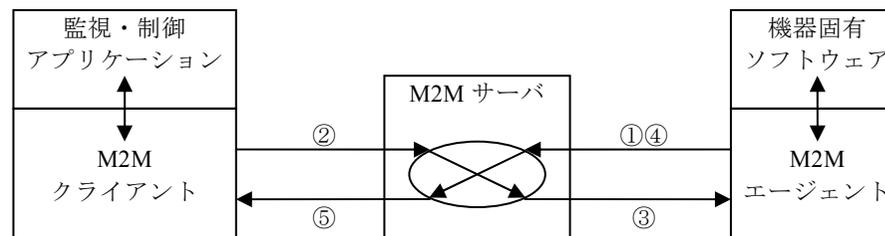


図2 Managed M2M システムの動作

- M2M サーバ：M2M クライアントと M2M エージェントの間の通信を中継する。各 M2M クライアントと M2M エージェントには固有の ID (M2M ID) が割り当てられており、この ID に基づいて中継処理を行う。インターネット上にアドレスを公開して、外部からアクセス可能でなければならないため、DMZ 等に設置する必要がある。

以下、Managed M2M システムの動作を図2に基づいて説明する。

- ① M2M エージェントが、M2M クライアントから要求が来ていないか確認する問い合わせを HTTP リクエストの形式で M2M サーバに送信する。M2M サーバは、M2M クライアントからの要求が一定時間（デフォルトで 30 秒）内に来なかった場合は、M2M エージェントに「要求なし」の応答を返す。
- ② アプリケーションからの要求により、M2M クライアントが機器への要求を HTTP リクエストの形式で M2M サーバに送信する。
- ③ M2M サーバは M2M エージェントからの問い合わせに対する応答として、機器への要求を M2M エージェントに送信する。
- ④ M2M エージェントは機器への要求を機器固有ソフトウェアで処理し、その結果を次の要求確認問い合わせとともに M2M サーバに送信する。
- ⑤ M2M サーバは M2M クライアントからの要求に対する応答として、処理結果を M2M クライアントに送信する。

以上のように、機器への要求と処理結果の送受信を、襷がけのような HTTP リクエスト送受信で実現しているため、Managed M2M システムには、以下の特長がある。

- ① 安全性：M2M クライアントも M2M エージェントも通常の HTTP リクエスト形式およびポート番号を使用してメッセージを送信しているため、ファイアウォールの内側に置くことができ、また外部にアドレスを公開する必要もない。
- ② 即時制御性：M2M クライアント側から見ると、送信した機器への要求を、M2M エージェントが即時に受信して処理することになる。

③ ネットワークキャリア非依存性:通信方式として HTTP のみを使用しているため、特殊なネットワーク機器は不要である。

なお、上記では HTTP で説明を行ったが、HTTPS (SSL 通信) も使用可能であり、これによって通信路上の安全性も確保される。

3. 最大同時接続数の評価

3.1 評価の目的

2章で述べたように、Managed M2Mシステムでは、M2Mエージェントが送信した HTTP リクエストを、M2Mサーバが一定時間保持してM2Mクライアントからの処理要求を待ち受けることによって、M2Mクライアント側から見た即時通信を実現している。このため、各M2MエージェントとM2Mサーバは、M2MサーバがM2Mエージェントに「要求なし」の応答を返し、M2MエージェントがM2Mサーバに次の問い合わせを送信するまでの短い期間を除いて、常時接続されていることになる。

一般的な Web サーバと Web ブラウザの関係においては、Web サーバからデータを送信した後は接続状態を保持する必要がないため、サービス可能な Web ブラウザ数は、同時接続可能な数より多くなる。一方、Managed M2M システムの場合は、監視・制御が可能な機器数 (=M2M エージェント数) は M2M サーバの同時接続可能数によって制限されることになる。

このため、1 台の M2M サーバに多数の M2M エージェントを接続して M2M サーバの負荷および M2M クライアントから見た応答時間を測定し、最大同時接続数を評価することとした。

3.2 評価システムの構成

図 3 に評価システムの構成、表 1 に各構成要素の諸元を示す。M2Mエージェントには、マルチスレッドで多数の問い合わせを発行する評価用のモジュールを用いた。

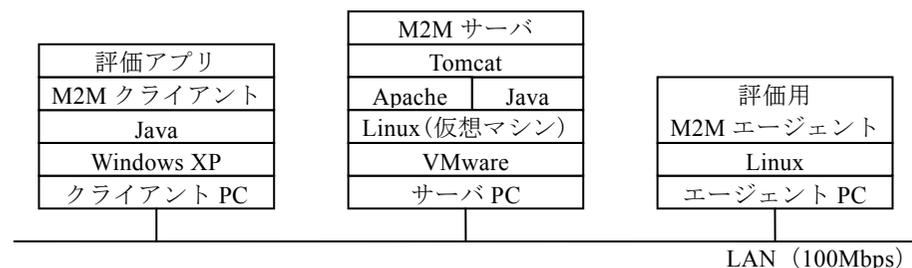


図 3 最大同時接続数評価システムの構成

表 1 最大同時接続数評価システムの諸元

構成要素	項目	諸元
サーバ	H/W	CPU : Xeon 4core 2.95GHz, メモリ : 10GB
	VMware	VMware Infrastructure 3.5
	仮想マシン	仮想 CPU 数 : 4, 仮想メモリ : 4GB
	Linux	CentOS 5.4 (32bit)
	Apache	Version 2.2.3
	Java	Java SE 1.6.0_17
	Tomcat	Version 6.0.20
エージェント	H/W	CPU : Pentium 4(HT) 2.80GHz, メモリ : 1GB
	Linux	Ubuntu 9.10
クライアント	H/W	CPU : Pentium 4(HT) 3.80GHz, メモリ : 3GB
	Windows	XP Professional SP3
	Java	Java SE 1.6.0_17

この評価では、M2M サーバの最大同時接続数を評価することが目的であるため、インターネットではなく LAN 上で評価を行った。また、M2M サーバと M2M エージェントおよび M2M クライアント間の通信には SSL を用いた。

なお、サーバ PC とエージェント PC では、多くのネットワークポートを同時に使用するため、Linux のパラメータを変更して、同時使用可能ポート数を増やしている。また、サーバ PC の Apache および Tomcat についても、種々の設定パラメータを変更して、同時に多数の HTTP/HTTPS リクエストを処理できるようにしている。

3.3 評価内容

評価用M2Mエージェントのスレッド数を 1, 500, 1000, 1500, …と 500 ずつ増やしていき、表 2 に示す項目の評価を行った。M2MサーバにおけるM2Mクライアントからの要求待ち時間は 30 秒とした。

表 2 最大同時接続数の評価内容

評価項目	評価方法
M2M クライアント側から見た応答性能	評価アプリで応答時間を測定して、5 回の平均値を求める。
M2M サーバの CPU 使用率	VMware パフォーマンスモニタの表示値。
M2M サーバのメモリ使用量	free コマンドの “-/+ buffers/cache” の used の値。

3.4 評価結果

図4に応答性能, 図5にCPU使用率, 図6にメモリ使用量の測定結果を示す.

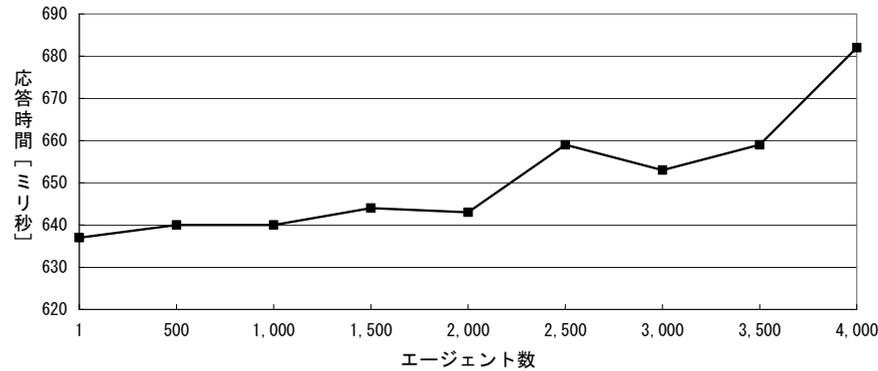


図4 エージェント数と応答性能

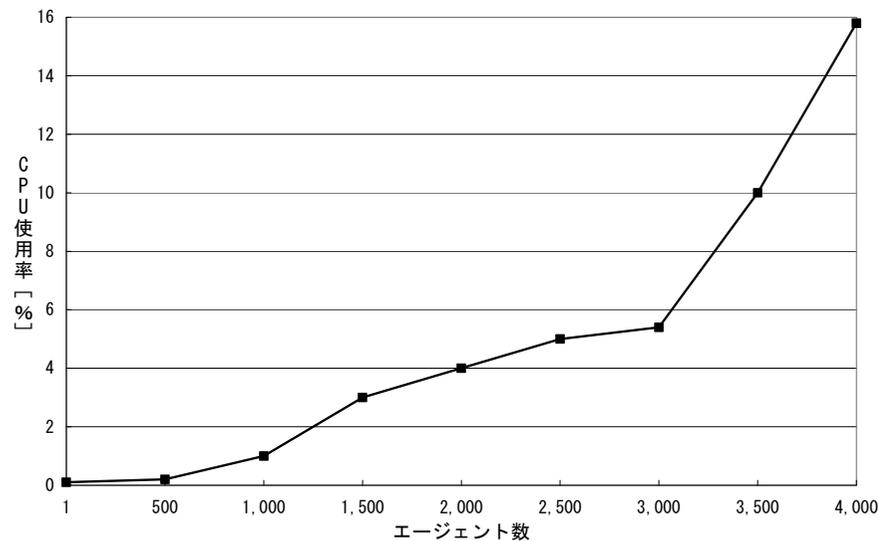


図5 エージェント数とCPU使用率

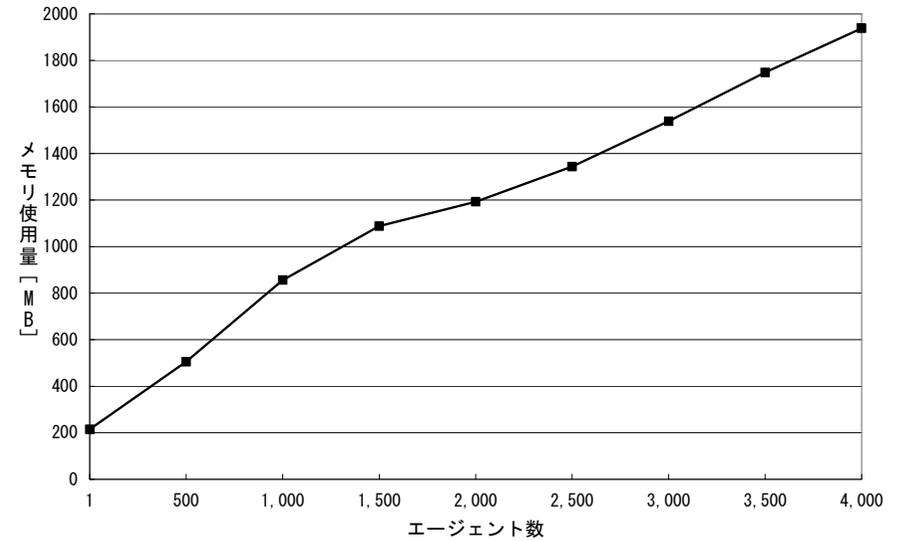


図6 エージェント数とメモリ使用量

応答時間については, エージェント数 2,000 まではほぼ横ばいで, 2,000 を超えると徐々に増大するが, エージェント数が 4,000 の場合でも 0.7 秒以下であり, 実用上, 問題ないことが確認できた. CPU 使用率については, エージェント数 3,000 を超えると上昇率が高くなるが, エージェント数 4,000 でも約 16%であり, M2M サーバの CPU 負荷は低いと言える. 一方, メモリ使用量については, エージェント数 4,000 で約 2GB であり, M2M サーバを動作させるには多量のメモリが必要であることがわかった.

なお, エージェント数が 4,000 を超えるとエージェント PC の CPU 使用率が急激に上昇し, M2M クライアントからの処理要求がタイムアウトになることもあったため, 今回の評価ではエージェント数 4,000 までの値を測定した.

4. データサイズと応答性能の評価

4.1 評価の目的

Managed M2M システム技術では, 現状, M2M クライアントから送信したコマンドも, それに対して M2M エージェントが送信する応答データも, 一度 M2M サーバのメモリ上に保持される. また, M2M サーバと M2M クライアント, M2M サーバと M2M エージェント間の送受信メッセージに XML を用いており, 機器からの応答データは

Base64 エンコードして XML 中に格納して送受信を行っている。したがって、送受信データ量が大きくなると、応答時間が増大することが予想される。

このため、M2M エージェントからの応答データサイズによって M2M クライアントから見た応答時間がどのように変化するかについて評価を行った。

4.2 評価システムの構成

図7に評価システムの構成、表3に各構成要素の諸元を示す。この評価では、M2M クライアントの処理中にはM2MエージェントはM2Mクライアントからの要求受信待ち状態、M2Mエージェントの処理中にはM2MクライアントはM2Mエージェントからの応答待ち状態で、同時に動作することがないため、M2MクライアントとM2Mエージェントを同一のPC上で動作させている。

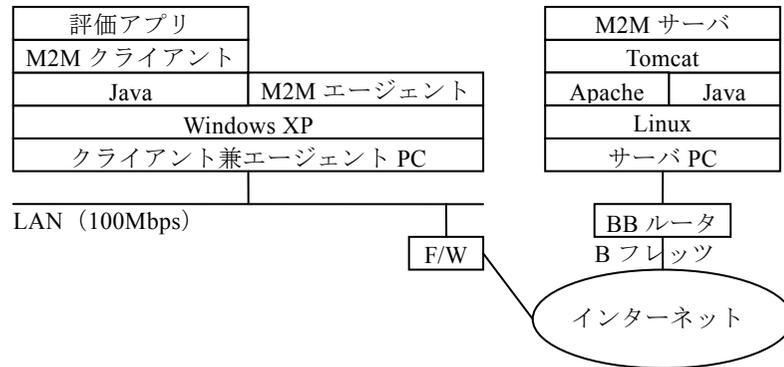


図7 データサイズと応答性能評価システムの構成

表3 データサイズと応答性能評価システムの諸元

構成要素	項目	諸元
サーバ	H/W	CPU : Xeon 3.80GHz, メモリ : 7.2GB
	Linux	Red Hat Enterprise Linux AS release 4 update 2
	Apache	Version 2.2.0
	Java	Java SE 1.5.0_06
	Tomcat	Version 5.5.16
クライアント兼エージェント	H/W	CPU : Pentium 4(HT) 3.80GHz, メモリ : 1GB
	Windows	XP Professional SP3
	Java	Java SE 1.6.0_17

4.3 評価内容

M2Mエージェントからの応答データサイズを、1 バイト、1MB、2MB、…と 1MB ずつ増やしていき、表4に示す項目の測定を行った。

表4 データサイズと応答性能の評価項目

項目	内容
M2M 処理	M2M クライアントが要求を送信してから、応答データを受け取るまでの時間
XML+Base64	受信した応答データ (XML) からデータを取り出して Base64 デコードするまでの時間
ファイル出力	データをファイル出力するまでの時間

4.4 評価結果

評価結果を図8に示す。

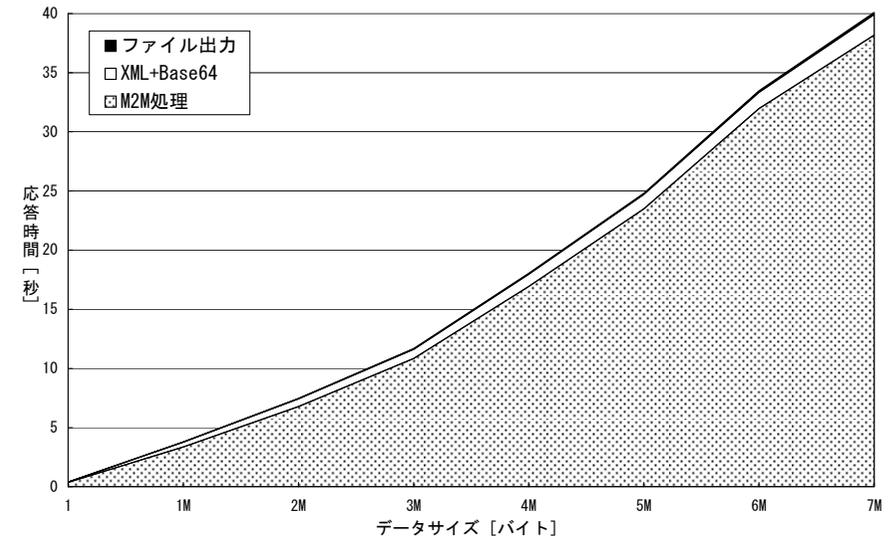


図8 データサイズと応答性能

上記のグラフに示すとおり、応答時間はデータサイズにほぼ比例している。また、応答時間の大半 (90~95%) は M2M 処理 (通信時間を含む) が占めており、XML や

ファイル出力の処理時間は、あまり大きくない。

5. 今後の課題

今回の評価で、M2M エージェント（すなわち監視・制御対象の機器）を同時に 4,000 接続しても性能的には問題ないことが確認できたが、評価用機器の拡充によって 4,000 以上のエージェント接続時の性能を測定する必要がある。

しかし、4,000 エージェントを接続した状態でメモリ使用量が約 2GB になっており、より多数の機器を接続すると正常に動作しない可能性がある。これを解決する手段として、通常の大規模な Web サーバシステムで用いられているロードバランサを用いることが考えられる。複数の M2M サーバに対する処理要求をロードバランサで振り分けることにより、大規模な Managed M2M システムの構築が可能となる。ただし、Managed M2M システムでは、M2M ID によって M2M エージェントとクライアントを識別しており、M2M エージェントが M2M クライアントからの要求を待ち受けている M2M サーバに対して、その M2M エージェントへの処理要求を送信する M2M クライアントが接続されなければならない(図 9)。これは、ロードバランサに送信する HTTP リクエストに M2M ID をパラメータとして付加し、ロードバランサの設定によって、同一の M2M ID パラメータを持つ HTTP リクエストが同一の M2M サーバに送信されるようにすることで解決できると考えられる。

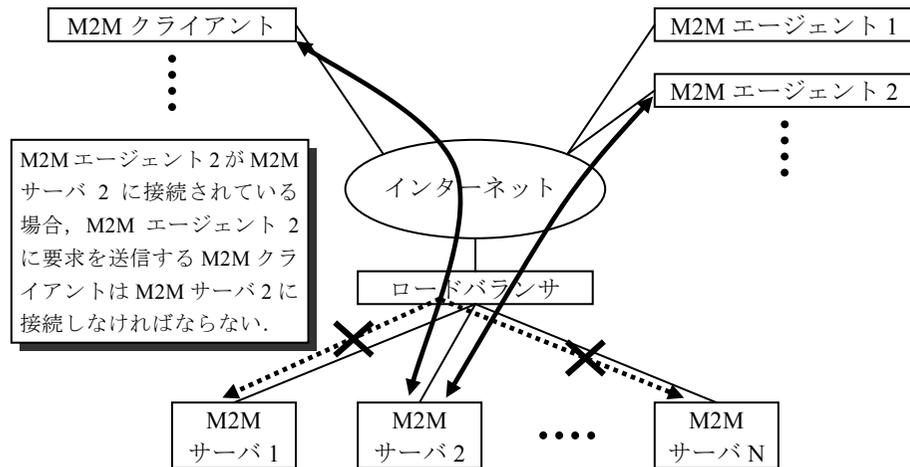


図 9 ロードバランサによる M2M サーバの負荷分散

また、大容量データの送受信ではデータサイズにほぼ比例した時間を要しており、用途によってはこれを短縮する必要がある。これを解決する方法としては、データのサイズが大きい場合は（一度に送受信を行うのではなく）複数の処理要求送信と応答データ送信に分割する方法や、M2M サーバと M2M エージェントおよび M2M サーバと M2M クライアントの送受信を同時に行う（M2M エージェントから応答データを受信しながら M2M クライアントにデータを送信する）方法などが考えられる。

6. おわりに

以上、Managed M2M システム技術の性能評価結果について述べた。性能評価の結果、機器数 4,000 程度までは、問題なく遠隔監視・制御が可能であることが確認できた。また、応答時間は送受信データサイズにほぼ比例することがわかった。

今後は、本技術を実際のシステムに適用して評価するとともに、「今後の課題」で述べた改良方法についても検討・実装を行っていく予定である。

参考文献

- 1) 金子洋介, 釜坂等, 道下学: Managed M2M システム技術(1)–複数の機器の統一的な遠隔制御と遠隔監視を実現–, 情報処理学会第 68 回全国大会(2006).
- 2) 釜坂等, 道下学, 金子洋介: Managed M2M システム技術(2)–インターネットを介した安全な即時制御を実現–, 情報処理学会第 68 回全国大会(2006).
- 3) 釜坂等, 金子洋介, 安田晃久: Managed M2M システム技術–多種の機器を効率的に一括制御するゲートウェイ機能–, 情報処理学会第 69 回全国大会(2007).