

3U-5

ミニMAPオブジェクトディクショナリシステムの性能評価

楠 和浩 宮内 直人 中川路 哲男 勝山 光太郎 水野 忠則

三菱電機(株) 情報電子研究所

1. はじめに

工場内の各種自動化機器の相互接続をマルチベンダ環境下で実現するための工場自動化用LAN (Local Area Network) の標準通信規約であるMAP (Manufacturing Automation Protocol) は、OSI (Open Systems Interconnection) 標準のサブセットとして、その標準化が進んでいる^[1]。MAPの応用層サービスの1つにネットワーク管理があり、その中でディレクトリサービスを規定している。

一方、MAPでは応答性を高めるために第3層(ネットワーク層)から、第6層(プレゼンテーション層)を除いたミニMAPを規定しており、応用層サービスとしてディレクトリサービスに対応したMAP独自のオブジェクトディクショナリシステム(Object Dictionary System)を用意している。

本稿では、ミニMAPオブジェクトディクショナリシステムの性能測定結果を報告する。

2. ミニMAPオブジェクトディクショナリシステムの概要

ミニMAPオブジェクトディクショナリシステムは、ミニMAPにおけるディレクトリサービスを提供するシステムである。ディレクトリサービスは、ネットワークの各資源を物理的な位置に関係なく名前で認識し、必要なアドレスを求める機能を持つ。

ミニMAPオブジェクトディクショナリシステムは、以下のものから構成される。

- (1) mDUA (mini-MAP Dictionary User Agent)
…オブジェクトディレクトリを利用するユーザを示す応用プロセス。
- (2) mDSA (mini-MAP Dictionary Service Agent)
…DIBを管理し、DIBから情報を読み出したりDIBの内容を更新したりする応用プロセス。
- (3) mDIB (mini-MAP Dictionary Information Base)
…アドレス情報を持つデータベース。
- (4) LDIB (Local Dictionary Information Base)
…DIBの部分的コピー。

mDIBはシンボル名とアドレスの対応表であり、mDSAにより管理される。各ステーションにはmDUAがあり、mDIBの部分的コピーであるLDIBが存在する。ユーザのアプリケーションプログラムは、mDUAを介してmDIBまたはLDIBからアドレスを得ることができる。

3. ミニMAPオブジェクトディクショナリシステムの機能と構成

現在までに我々は、MAP 3.0に基づいたミニMAPオブジェクトディクショナリをEWS上に第2層以下のシミュレータと同時に開発を行った^[2]。図1に今回の性能測定の対象となるミニMAPオブジェクトディクショナリシステムの構成を示す。

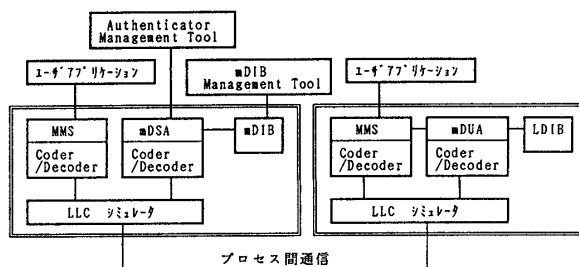


図1 ミニMAPオブジェクトディクショナリシステムの構成

本システムで実現したオペレーションは以下の通りである。

- (1) 読みだし (QueryDibEntry)
 - (2) 追加 (AddDibEntry)
 - (3) 削除 (DeleteDibEntry)
 - (4) 変更 (ChangePhysicalAddress)
 - (5) 更新通知 (UpdateDibBroadcast)
 - (6) mDSAのロケーション要求 (WhereIsMDSA)
 - (7) mDSAのロケーション通知 (ThisIsMDSA)
- このうち(1)~(4)はシングルポイントオペレーション、(5)~(7)はブロードキャストオペレーションである。

4. 性能測定の目的と測定方法

4.1 性能測定の目的

今回の性能測定の目的は、次の3点である。

- (1) ミニMAPオブジェクトディクショナリのプロトタイプとしてどの程度の性能が得られるかを知る。
- (2) ミニMAPオブジェクトディクショナリの各オペレーションの特徴を求める。
- (3) 性能面でネックとなる箇所を求める。

4.2 性能測定方法

今回、ミニMAPオブジェクトディクショナリシステムの測定にはEWS (68020 20MHZ)を用いた。また、処理時間の測定のためには、UNIXの"TIMES"コマンドを用いた。

今回の測定では、mDSA側で

- ① 受信したPDUの分解にかかる平均CPUタイム
 - ② mDIBに対するアクセスにかかる平均CPUタイム
 - ③ mDSAからmDUAに結果を返すためのPDUの生成にかかる平均CPUタイム
- について性能測定を行った。

5. 測定結果と考察

例として、QueryDibEntryオペレーションとAddDibEntryオペレーションに関する測定結果を以下に示す。

(1) QueryDibEntryオペレーション

表1、図2、表2にそれぞれの測定結果を示す。

表1 PDUの分解にかかる平均CPUタイム

| | 平均CPU使用時間 |
|--------|------------|
| PDUの分解 | 5.305 msec |

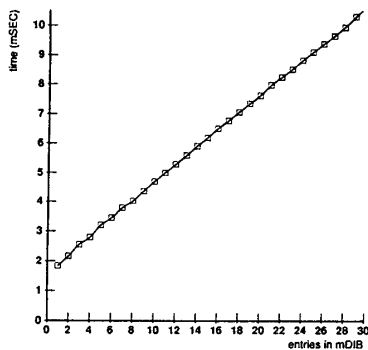


図2 mDIBへのアクセスにかかる平均CPUタイム (QueryDibEntryオペレーション)

表2 mDSAからmDUAに結果を返すときのPDUの生成にかかる平均CPUタイム

| | 平均CPU使用時間 |
|--------|-------------|
| PDUの生成 | 14.159 msec |

(2) AddDibEntryオペレーション

表3、図3、表4にそれぞれの測定結果を示す。

表3 PDUの分解にかかる平均CPUタイム

| | 平均CPU使用時間 |
|--------|-------------|
| PDUの分解 | 14.978 msec |

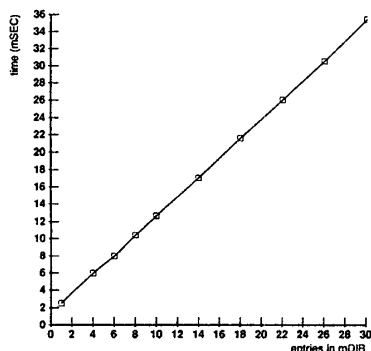


図3 mDIBへのアクセスにかかる平均CPUタイム (AddDibEntryオペレーション)

表2 mDSAからmDUAに結果を返すときのPDUの生成にかかる平均CPUタイム

| | 平均CPU使用時間 |
|--------|------------|
| PDUの生成 | 6.077 msec |

プロトタイプとしてのミニMAPオブジェクトディクショナリシステムの性能は、結果からわかるように最も頻繁に実行されると考えられるQueryDibEntryオペレーションの場合にはmDSA側での処理に約24msec (mDIBに登録されているエントリ数が10の場合) かかり、AddDibEntryオペレーションでは約34msecかかる。

例に示した2つのオペレーション (QueryDibEntryオペレーションとAddDibEntryオペレーション) の特徴としては次のことが挙げられる。

QueryDibEntryオペレーションでは、mDSAに送られてきたPDUに比べてmDSAからmDUAに送り返される結果のPDUの方が、扱うデータ量が多いので、mDSAでは当然PDUの生成にかかるPDU時間の方が大きい。AddDibEntryオペレーションでは、扱うデータ量の関係はQueryDibEntryオペレーションの場合とは逆なので結果も逆になっている。

PDUの分解とPDUの生成にかかるCPU時間の和はQueryDibEntryオペレーションとAddDibEntryオペレーションでは、ほぼ同じであるがmDIBにアクセスするのにかかるCPU時間がAddDibEntryオペレーションの方が多い。しかもmDIB内のエントリ数が多いほどその差は大きくなっている。この原因は次のように考えられる。AddDibEntryオペレーションではmDIBにエントリを登録する際に、①登録しようとするエントリのシンボル名と同じシンボル名のエントリが既に登録されていないか、②登録しようとするエントリのLSAPアドレスと同じLSAPアドレスが既に登録されていないか、というチェックを行う。このチェックにかかる時間のためにQueryDibEntryオペレーションよりもAddDibEntryオペレーションの方がmDIBにアクセスするのにかかるCPU時間が多くっており、さらにmDIBに登録されているエントリ数が多いほどその差が大きくなっていると考えられる。

システムの性能を改善するために、以下のような対策を考えている。

- ① 現在ファイルとなっているmDIBをmDSAの立ち上げ時にメモリ上にもってくる。
- ② QueryDibEntryオペレーションは、メモリ上のmDIBへのアクセスとし、AddDibEntryオペレーション、DeleteDibEntryオペレーション、およびChangePhysicalAddressオペレーションはメモリ上のmDIBでチェックを行い、検索時間を短縮する。

6. おわりに

ミニMAPオブジェクトディクショナリの性能を評価した。今後は今回の性能評価に基づいた改善を行う予定である。

参考文献

- [1] General Motors Corporation : MANUFACTURING AUTOMATION PROTOCOL SPECIFICATION Version3.0 , C-13 (1987)
- [2] 楠他 : ミニMAPオブジェクトディクショナリシステムの实现 , 電子情報通信学会春季全国大会 (1989年) B-458 , (1989)