

マルチプロトコル対応インターネット連携通信ミドルウェアMINTの設計と実装

2 R - 4

今井 功 祢宜 知孝 楠 和浩 下間 芳樹
三菱電機(株)情報技術総合研究所

1. はじめに

これまでLAN(Local Area Network)環境で使用してきたFA(Factory Automation)機器に対する保守・監視アプリケーションを、インターネット環境に対応させる手段として、我々は通信ミドルウェアMINT(Mitsubishi InterNet Translator) Version1.0(以下、MINT1.0)を開発した[1]。MINT1.0では、LAN環境における通信プロトコルとしてUDP/IPプロトコルをサポートしたが、今回、新たにシリアル通信をサポートするための検討を行った。

本稿では、インターネット連携通信ミドルウェアMINTのマルチプロトコル対応における課題点とその解決策について報告する。

2. MINT1.0概要

図1に、MINTの内部構成を示す。本ソフトウェアは、インターネット経由による既存ソフトウェアとFA機器間の通信処理を支援する。

保守端末上に実装したMINT Client(以下、MINT-C)は、アプリケーション連携処理モジュール、HTTPラッピングモジュール、及びインターネット連携処理モジュールの3つの処理モジュールから構成される。また中継機上に実装したMINT Server(以下、MINT-S)は、インターネット連携処理モジュール、HTTPラッピングモジュール、及びインターネット連携処理モジュールの3つの処理モジュールから構成される。

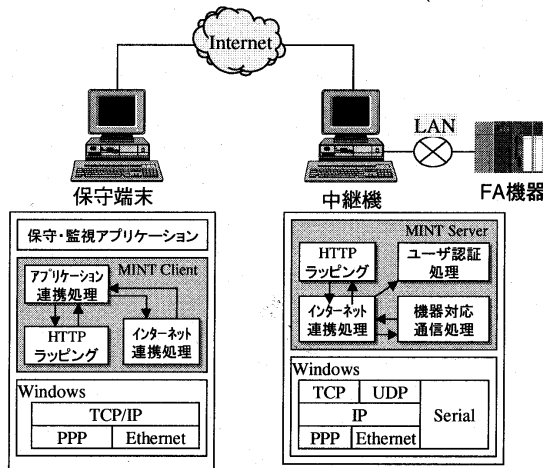


図1 MINTソフトウェア構成

A Design and Implementation of the Multiple Protocol Supported Middleware via Internet - MINT.
Isao Imai, Tomonori Negi, Kazuhiro Kusunoki, and Yoshiaki Shimotsuma.
MITSUBISHI ELECTRIC CORPORATION
5-1-1 Ofuna, Kamakura, Kanagawa. 247-8501, Japan.

T-S)は、インターネット連携処理モジュール、HTTPラッピングモジュール、ユーザ認証処理モジュール、及び機器対応通信処理モジュールから構成される。尚、図中の矢印は、モジュール間のデータの流れを意味する。

本ソフトウェアでは、既存ソフトウェアの汎用性及び軽負荷化を考慮し、インタフェースを出来る限り簡素化する事を心掛けた。アプリケーション連携処理モジュールは、上位に位置する保守・監視アプリケーションに対し、以下のインタフェースを提供する。

表 1 インタフェース一覧

インタフェース名	機能
MintOpen	MINT-C/MINT-S間のコネクションを確立する。
MintClose	MINT-C/MINT-S間のコネクションを開放する。
Initialize	パスワードによるユーザ認証処理を行う。認証後、機器構成情報を取得する。
SimpleInitialize	パスワード認証処理のみを行う。
DataSend	FA機器に対し要求データを送信する。
DataReceive	FA機器から応答データを受信する。
SetTimer	データ送受信時のタイムアウト時間を設定する。
TraceInit	実行時に発生するイベント内容や通信データログをファイルに出力する。

MINT-Cのインターネット連携処理モジュールは、アプリケーション上で生成されたFA機器が利用する通信プロトコルに対応した通信データをTCP/IPヘッダに隠蔽し、インターネットを介してMINT-Sに送信する。MINT-S側のインターネット連携処理モジュールは、受信データから通信データを取り出し、機器対応通信処理モジュールを通じてFA機器に送信する。機器対応通信処理モジュールは、各種通信プロトコルに応じた処理機能を内部に持ち、インターネット連携処理モジュールに対し通信プロトコル非依存なインタフェースを提供する。

また、MINT-Cは、アプリケーションの処理を効率よく進める為に、MINT-Sに送信データが到着したことを認識した時点で制御をアプリケーションに返す。

3. マルチプロトコル対応における課題点

FA機器対応通信プロトコルとしてシリアル通信のサ

ポートを検討した結果、現状のアーキテクチャには、以下の課題点がある事が分かった。

課題(1) MINT-SとFA機器間の通信回線制御

表1に示す通り、MINT1.0が中継機とFA機器間の通信処理として提供しているインタフェースは、データの送受信処理のみである。一方、シリアル通信は、直接通信回線の信号制御を行う必要があるため、MINT経由でシリアル回線制御を行うための新たな仕組みが必要となる。

また、シリアル通信では、シリアル回線に対する信号制御や状態取得などの処理が頻繁に行われる。これらの処理要求が、逐次インターネット経由で発生すると、性能劣化となる事が予想される。

課題(2) データ送信時における回線使用タイミング

MINT1.0では、MINT-SとFA機器間の通信処理として、データ送信用とデータ受信用にスレッドを分けた並行処理方式を採用している。シリアル通信では、COMポートを同時に複数のスレッド間で共有することができず、いずれか一方のスレッドしかCOMポートを使用することが出来ない。よって、MINT-Sがデータ送受信処理を行う場合、スレッド間でCOMポートを使用するタイミングを調整する必要がある。

課題(3) シリアル通信とTCP/IPにおける通信性能差

シリアル回線を利用した通信処理は、TCP/IPプロトコルに比べ通信処理性能が劣る。そのため、MINT-CがFA機器に対して要求データの送信直後に応答データの受信要求を行うと、FA機器からの応答データが到着しない事により、MINT-Sが受信できない場合が起こり得る。

4. 実現方式

4.1. 課題(1)に対する解決策

本方式では、MINT-SとFA機器間の通信回線制御を実現するため、MINT-C及びMINT-S内のインターネット連携処理モジュール間に、図2に示す様に

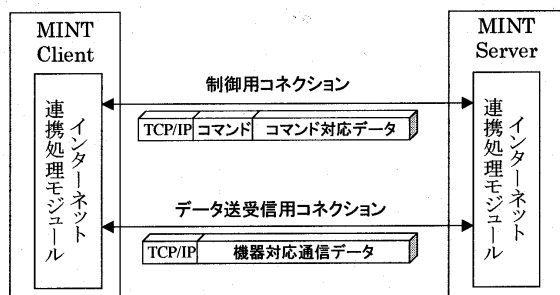


図2 用途別通信コネクション

制御用とデータ送受信用の2種類の通信コネクションを用意した。制御用コネクションは、FA機器に対する制御命令（シリアル回線のON/OFF制御、状態取得、並びに伝送速度設定等）をコマンドとして発行し、その処理結果を受信するために利用する。一方、データ送受信用コネクションは、FA機器対応通信データの送受信処理に利用する。

また、シリアル回線に対する信号制御には、決まった処理手順が存在する場合がある。本方式では、MINT-SとFA機器間における制御命令を一括して処理するプロキシ実行機能を用意した。これにより、インターネット間の通信回数が削減され、結果として応答性能を向上することができた。MINT-Cからは、MINT-Sに対し一括処理要求メッセージを発行する。MINT-Sでは、これに対応するプロキシ機能を実行し、結果をMINT-Cに返信する。

4.2. 課題(2)に対する解決策

UDP/IP通信プロトコルでは、MINT-Sの制御機器対応通信処理モジュールにおいて、受信用スレッドが常時データ受信待ち状態にある。これに対し、シリアル通信の場合は、MINT-Cから制御用コネクションを通じて受信要求メッセージを受けた時点で、MINT-Sが応答データの受信待ち状態となる方式とした。FA機器から受信した応答データは、データ送受信コネクションを通じてMINT-Cに返信される。

4.3. 課題(3)に対する解決策

本方式では、通信プロトコル間の処理性能差により応答データを取得する事が出来ない場合の対策として、MINT-Cから自動的に要求メッセージを再送する機能を用意した。MINT-Sは、MINT-Cから再度受信要求を受けることにより、FA機器から応答データを取得し、MINT-Cに対し返信する。

5. おわりに

本稿では、LAN環境における通信プロトコルを複数対応可能とするためのMINTの実現方式について述べた。今後は、他通信プロトコル適用によるレパトリ拡充及び通信データに関するセキュリティ強化を実施する予定である。

参考文献

- [1] 今井他：インターネットを利用した遠隔アクセスシステムの設計と実装，マルチメディア、分散、協調とモバイル(DICOMO'99)シンポジウム論文集，情報処理学会，pp.279-284，1999。