

WEB技術を用いたプラント遠隔監視システムの 風力発電設備への適用

江 藤 春 日***・深 堀 潔**
松 尾 博 文***・黒 川 不二雄***

Remote monitoring system with WEB technology for a Wind turbine generation plant

by

Haruhi ETO***, Kiyoshi FUKAHORI**
Hirofumi MATSUO*** and Fujio KUROKAWA***

Recently, wind farms, the fastest-growing energy source in the world, has been built as an environmentally friendly source of electricity. In many cases, wind farms are located far from residential area due to their noise, wind availability, wind speeds and local weather conditions. From the economical and environmental point of view, it should be considered to reduce experts for operation and maintenance of wind turbine. For this purpose, the remote monitoring system which provides the operation and supervisory functions for wind farms has been required. This paper presents that the local network for the large-scale wind farm is realized and the multi-thread is effective to improve the total through-put of communication between server and the wind-turbines. At first the feature of the remote monitoring system is reviewed. Next, the procedure to derive the optimum number of thread is presented. Finally it is demonstrated that the required performance such as the total through-put of communication with the number of thread is satisfied in the 100 nodes network, which is derived by the proposed procedure.

1. はじめに

平成10年12月に開催された気候変動枠組条約第3回締約国会議(COP3)合意を受けて、わが国でも環境調和型エネルギー需給構造の構築への取組が進んでいる。その中で、風力発電は重要な非化石新エネルギーとして位置付けられ、1997年末では1.8万kW¹⁾であった電力供給能力を2010年には約170倍の300万kWとする政府計画で、風力発電設備の建設は今後急速に増加する事が予想される²⁾。一方、風力発電設備は風況や周辺地域への騒音への配慮からわが国では交通や通信の不便な辺鄙な場所に建設されることが多

い³⁾。設備の安定的な稼動を維持するためには、運転状況や設備の異常を常に監視することが望ましいが、運転監視の要員を風車の近くに常駐させることはコスト及び労働環境の面から難しい。そこで、遠隔地で風力発電設備を監視し起動停止などの運転ができ、故障や事故の情報を収集できる装置が求められてきている。

最近のインターネット通信技術は有線、無線の両方で急速に進歩し安価なネットワークの普及が進んでおり、風車のサイトと遠隔地をつないだ監視システムにも既に使われている。しかしながら、従来の遠隔監視

平成13年10月26日受理

*大学院海洋生産科学研究科 (Graduate school of Marine Science and Engineering)

**長菱制御システム株式会社 (Choryo Control System Co., Ltd.)

***大学院生産科学研究科 (Graduate school of Science and Engineering)

システムは、1台から数台規模の比較的小規模の風車サイトを対象としており、各風車のデータ収集にはプロセス入出力やRS-485などのシリアル通信が使われてきた。また、近年、更なる省力化や運転監視内容の充実のため、通信のデータ量が増大し更新周期の短縮も求められており、従来のインターフェースでは100台規模の大規模ウィンドファームの風車を対象とした場合、通信性能の面でシステムの要求にこたえられないという問題点があった。さらに、風力発電設備においては雷害³⁾が問題となってきており、風車サイトにおける通信線の光ファイバー化の要求も多く、従来のインターフェースに代わるネットワークが求められている。

そこで、従来のインターフェースの問題点に対する対策として、風車サイトのローカルネットワークに、産業用には不向きとされてきたイーサネットを採用し、大規模風力発電設備の遠隔監視システムを実現している⁴⁾。この様な、大規模風車サイトのネットワークにおいては接続された多数の風車のデータを決められた時間内に更新できるだけのスループットを確保することが最も重要である。このためには、ソフトウェアのオーバーヘッドを可能な限り小さくすることが必要であり、その手段として、プロセスの並行処理の1つであるマルチスレッド方式を用いることにする。ここで、スレッドとは独立に動作するアプリケーションプログラムの実行単位で、オペレーションシステムにおけるタスクの概念に似たものである。⁵⁾

本論文では、このスレッド数とネットワークのトータルスループットの関係および、それに基づく最適なスレッド数を示し、大規模風車サイトにおける、ローカルネットワークの有効性および設計手順を明らかにする。

まず、風車の遠隔監視システムの構成と機能について述べ、従来のインターフェースの問題点、および、ローカルネットワークにイーサネットを採用した理由を示す。次に、通信データ量と接続する風車の数に対する最適なスレッドの数の決め方を示し、マルチスレッドを用いた場合に100台の風車と1秒以内に通信するために必要なトータルスループット、1.12Mbps⁶⁾を確保するための最適なスレッド数を求める。最後に実際の通信におけるスループットを測定し評価する。

2. 風力発電設備監視システムの構成と機能

2.1 システムの構成

風力発電設備の制御・監視システムは図1のシステム構成図に示すように、個々の風車制御装置と、その

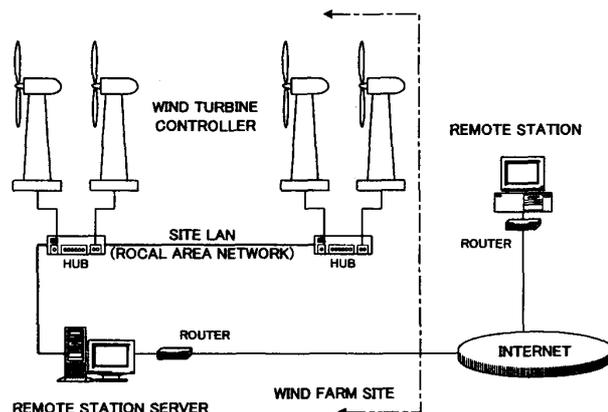


Fig.1 System configuration of proposed remote monitoring system for wind farm.

データを収集し外部の監視装置にデータを配信するデータロガー（リモートステーションサーバ）、外部監視装置（リモートステーション）及びそれらを接続するネットワークからなっている。ここで、風車制御装置は風速に対応した風車ブレードの角度制御や、風向に合わせたプロペラの方向制御などを行う組み込み型コンピュータ⁶⁾である。個別風車周りの風速・風向・気温などの環境状況や、発電電力や周波数など運転状態のデータ入力処理も行う。データロガーに対し通信を介して個別風車のデータを送信するとともに、データロガーを介して送られた風車の操作指令に従って風車の起動停止を制御する。

データロガー（リモートステーションサーバ）は風車の監視データを収録するとともに、インターネットへの接続のためのWEBサーバとして機能する。機能の概要については2.2.2で述べる。

外部監視装置（リモートステーション）はリモートステーションサーバとネットワークで接続され、風車の運転状態、風速・風向・気温などの環境状況、故障や事故の状態表示、発電量などの運転記録などをリモートステーションサーバから取り出してCRT画面上に表示する。個別の風車に対し起動や停止の操作を行う事もできる。

2.2 遠隔監視システムのソフトウェアの構成および機能

本章では遠隔監視システムのソフトウェアの構成と主な機能について簡単に紹介する。

図1のシステム構成図において、リモートステーションサーバとリモートステーションを遠隔監視システムと位置付ける。

また、リモートステーションサーバはリモートステ

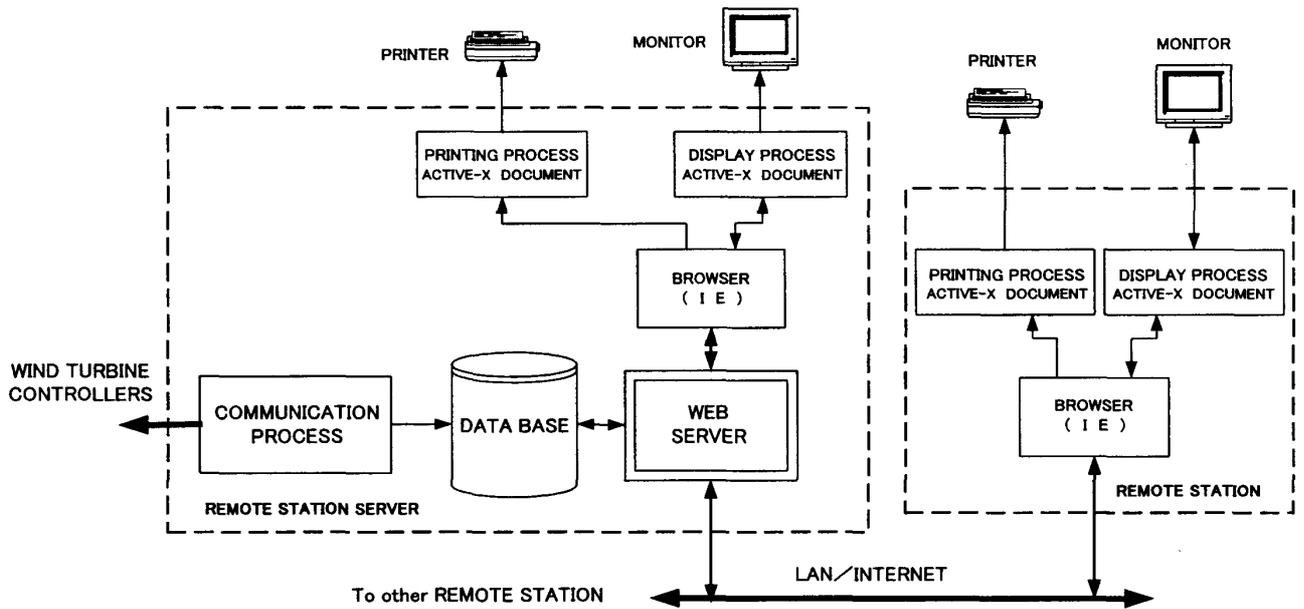


Fig. 2 Software configuration of proposed remote monitoring system.

ーションの機能を包含しているため、これ以降、遠隔監視システムとしての記述はリモートステーションサーバをさす。

2.2.1 ソフトウェアの構成

図2にリモートステーションサーバ及びリモートステーションのソフトウェアの構成を示す。

リモートステーションサーバはオペレーティングシステム(OS)にWindows-NTを使用し、その上にWEBサーバ、データベースおよび風車ネットワーク通信プログラムを構成している。リモートステーションサーバ単独でも監視のマンマシンインターフェースが機能するように、リモートステーションと同じブラウザを組み込み、ソケット⁵⁾でWEBサーバと通信させている。リモートステーションにはWEBブラウザが必要でOSとしてはWindows-NTを推奨している。また、WWWの技術体系であるActiveX⁷⁾を採用し、WEB画面のアプリケーション構築において、表現の自由度の向上や、遠隔地にあるシステムのActiveXコントロール部品の保守性の向上を図っている。

図2に示すリモートステーションサーバのソフトウェア構成の中で風車ネットワーク通信プログラムはネットワーク上の個別の風車制御装置と通信し、データを収集するプロセスである。

2.2.2 リモートステーションサーバ(データロガー装置)の機能

リモートステーションサーバの主要な機能として、データ収集機能、表示機能、WEBサーバ機能がある。それぞれの機能の概要を表1に示す。風車設備の性能改善や故障時の原因解析のためには詳細なデータの収集が必要であり、そのためにネットワーク上の全ての風車からのデータを1秒以内に更新することが要求されている。

3. 風車サイトにおけるローカルネットワークの適用

本システムではローカルネットワーク(サイトLAN)にイーサネットを採用した。本章ではイーサネットを採用した理由を述べるとともに、多数の風車を接続してデータを収集する場合の通信の総合的なスループットが通信ソフトウェアを並列に処理すること、すなわち、マルチスレッドを使用することで向上できることを示し、通信するデータ量と接続する風車の数に対する最適なスレッドの数を決定する方法を示す。ここで、スレッドとは独立に動作するアプリケーションプログラムの実行単位でオペレーションシステムにおけるタスクの概念に似たものである。

3.1 大規模ウィンドファームにおけるシリアル通信適用の問題点

従来、風車の監視システムでは、サイト(ローカル)

Table 1 Function of remote station server.

データ収集機能	個別の風車制御装置と通信し制御装置が収集した瞬時値データおよび、制御装置が検出したアラームデータを1秒以内の周期で収集する。
	収集したデータを基に発電設備の運転記録として、日報及び月報の形式で編集し、個別の風力発電設備の性能計算を行う。
表示機能 (WEBブラウザ表示機能)	一画面でウィンドファーム全体100台の風車の運転状態を一括して監視するオーバービュー画面、ファーム全体の風況、個別の風車の風速と発電量の数値や、異常発生中や停止中など風車の状態を色で一覧表表示する。
	個別の風車に関し、風車回転数、発電機出力、力率、風況、装置各部の温度、油圧などの詳細なデータを表示する画面。
	サブステーションのON,OFF,TRIP状態を色で区別し、電圧、力率、無効電力、送電電力、受電電力などの状態量を数値表示するサブステーション系統表示画面。
	発生した警報をリアルタイムで表示するとともに過去に発生した警報の履歴や警報ごとの発生の積算回数などを表示する警報表示画面。
	故障発生時に各風車が収集し保存しているデータを時系列で並べて表示する画面。故障原因の解析に有効。
WEBサーバ機能	各風車から通信で取り込まれ、集約されたデータを、WEB端末(リモートステーション)からの要求に応じて発信する機能。リモートステーションからのアクセスをパスワードで区別し、不正アクセスを防止すると同時に、予め設定されたリモートステーション以外からのアクセスに対してはパスワードが合致しても応答を拒否するファイヤーウォール機能を備えている。
	予め規定されたイベントが発生したとき、E-メールを自動で発信する機能。

に置かれたデータロガーと呼ばれるコンピュータがアナログ信号、接点信号による入力や、RS-422、RS-485などの平衡型シリアル通信を用いて、風車の監視データを収集し外部の表示監視装置がデータロガーと通信して監視画面などで表示する方法が取られてきた。しかし、プロセス入出力を用いる方法はハードウェアのコストが大きく近年ほとんど採用されていない。

RS-422や485はインターフェース回路や通信線が安価で、また、マルチドロップ接続により1つの通信セ

グメント上に複数台のノードが接続出来るので、10台未満程度の小規模ウィンドファームにおいては現在でも主要な手段である。しかし、2章で示したように監視項目が増え、風車制御装置とサーバ間の通信データサイズが大きくなってきた今日、数十台以上の大規模風力発電設備において、シリアル通信を用いる構成では、通信性能とネットワーク媒体の両面でネットワークの必要性能を満足できないという問題点がある。

まず、通信の性能に関して述べる。RS-422シリアル通信ではドライバー回路の規格の制約から、1つの通信セグメントに接続できるノード数は最大10程度(RS-485では32)であり、10台を越える風車を接続するためには、データロガー側に風車10台毎に1つずつ通信インターフェースが必要となる。また、風車が大型化するにつれて風車間の距離も大きくなってきており、600kWから1MW級の大型風車の場合、100~120m間隔で建てられるため、大規模ウィンドファームにおいては通信線路は数kmに及ぶ。この場合、RS-422や485では最大の伝送距離が1.2km以下という制約があり、伝送線の長さによって通信レートにも制限がある。仮に通信線路を1kmとした場合、最大の通信レートは120kbpsである。風車1台を監視するために必要なデータを1397バイトとしたとき100台の風車を監視するためには8.8秒以下の周期には出来ない。

次に、ネットワークの媒体について述べる。シリアル通信でネットワークを組む場合のトポロジーを図3に示す。風車はその目的から年間を通して風の強い場所に設置されるため、周辺は大気の状態が不安定で落雷が多い。近年、雷害対策として通信ラインを光ファイバーにする要求が増えてきているが光ファイバーではマルチドロップが出来ないため図3の構成は適用できず、光ファイバー適用のためには全てのノードとのスター接続になる。スター接続は非常に長い光ファイバーを必要とするため、媒体のコストが大きいという問題があり現実のシステムにおいては採用されない。

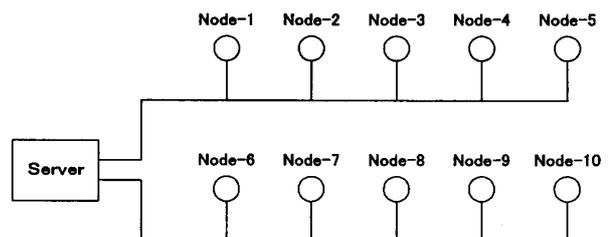


Fig. 3 A topology of the serial communication network.

3.2 サイト LAN へのイーサネットの適用

3.1 で述べたシリアル通信の問題点の対策としてデータロガーと風車制御装置との接続に、イーサネットによるローカルネットワークを適用することが考えられる。従来、イーサネットは業務系の通信には使えるが、産業用には不向きであるとされてきた。その理由はイーサネットの構造上発生する通信データの衝突による信頼性の低下とされている。しかし、データの衝突がシステムの性能に影響するような場合でも、衝突の問題はスイッチによって解決できる。従来高価だったスイッチも最近ではネットワークのハードウェアコストとしては問題とされないくらい安価になってきており、データ衝突による信頼性低下は解消されたとと言える。また、イーサネットの規格 IEEE802.3 で規定されている物理層、データリンク層の構造は産業用に必要な機能を全て備えており、関連ソフトウェアが充実し、周辺ハードウェアが非常に安価に手に入ることを加味すると現状最も使いやすいネットワークであると考えられる。

イーサネットでローカルネットワークを構築する場合、媒体に光ファイバーを適用する場合においてもハブを利用してトポロジーが柔軟に構成でき、データロガー側には1つの通信インターフェースを持つだけで10Mbpsの通信レートが実現できる。ローカルネットワークを介して個々の風車の情報を1台の共通サーバに集約し、そのサーバをWEBに接続する構成とすれば、遠隔の監視装置とデータロガーをWEB接続する回線のダイヤルアップルータやファイヤーウォールなどの設備が1セットで済むため外部との接続コストを少なくでき、合わせてセキュリティの確保が容易である。

以上のような理由で本システムではサイト LAN にイーサネットを採用した。

3.3 通信レートの選定

今日、イーサネットは通信レートとして100Mbpsまたは10Mbps、或いはその両方を併用することができる。大規模な風車のサイトはネットワークの総延長が数十kmに及ぶものもあり、通信媒体の選択の容易さやコストを考慮すると必要以上に高速にすると逆に不利になる。

2章で述べた風車1台を監視する機能に必要なデータは最大で1397バイトであり、100台の風車のデータを1秒毎に更新するとした時、1.12Mbps以上の通信レートがあればよい。従って、10Mbpsの通信レートがあると十分なので、システムの通信レートには10Mbpsを採用した。

3.4 サイト LAN におけるアプリケーションを含めた通信の性能とマルチスレッドの採用

風力発電の監視用ネットワークにおいては、データの確実な受け渡しと、通信周期の保証が求められるため、ポーリングによる送受信が一般的であり本論文でもポーリングによる通信について検討した。ここで、システムとしての通信の性能はネットワークのビットレートだけでは決まらず、アプリケーションでのオーバーヘッドも含めて評価しなければならない。サーバが一定の時間内に送受信できるパケットの数には送受信するデータの長さ、サーバ側のアプリケーションオーバーヘッド、風車側のアプリケーションオーバーヘッドが関わっている。

図4にポーリングによる1回の送受信モデルのタイミングチャートを示す。

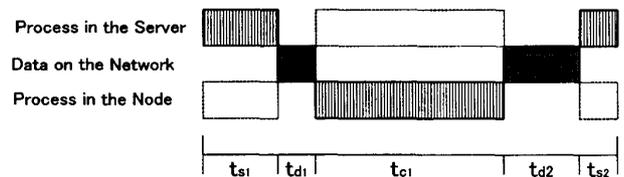


Fig. 4 Timing chart of a communication sequence.

図4において、 ts_1 はサーバがポーリングのパケットを準備する時間、 ts_2 はサーバがノードから受信したデータを処理する時間、 tc_1 はノード側がサーバからのデータ要求を処理し送信するデータパケットを準備する時間、 td_1 はサーバからのポーリングパケットが回線上を流れている時間、 td_2 はノードからのデータが回線上を流れている時間である。

ts_1 、 ts_2 は使用するコンピュータの性能と処理プログラムの作り方で決まる。また、 $td_1 + td_2$ はパケットのサイズで決まる。1回のポーリングによるデータ送受信にかかる時間 T_1 は

$$T_1 = ts_1 + td_1 + tc_1 + td_2 + ts_2 \quad (1)$$

であり、N台のノードとポーリングによる通信を行った場合、全てのデータの送受信にかかる時間 T_i は

$$T_i = nT_1 \quad (2)$$

となる。ここで、要求される更新周期がp秒のとき、 $T_i > p$ であれば、全てのノードのデータを定められた期間内で更新することは出来ない。また、一般にアプリケーションのオーバーヘッドの時間 $ts_1 + tc_1 + ts_2$ はデータが回線を流れている時間 $td_1 + td_2$ に対し何倍も大きい場合が多く、アプリケーションの構造を工夫してシステムとしてのスループットの向上を図る必要がある。

図4から容易にわかるように tc_1 は回線上の空き時間とみなせる。この空いた時間にもデータを流せるようにするため、サーバ側に並列処理のプログラムを導入し、複数のノードと非同期で並列に通信をすることによって、平均スループットを向上させることができる。並列処理のために、マルチスレッドを採用した。

3.5 スレッド数の決定方法

ここでは、スレッド数を決定するための手順を示す。

スレッドの数を増やせば回線に余裕がある限りスループットを高めることが出来るが、スレッドはそれぞれ、通信バッファを含めて大きなメモリを使用するのでスレッドの個数は通信のスループットがシステムの要求を満たす範囲で最小にとどめるべきである。

(1)式で、 tc_1 はノード側アプリケーションで要する時間であり、その間はサーバ側アプリケーションは動作可能である。回線上のデータの衝突を無視でき、また、各スレッドがスレッド間の干渉無しに動作できる理想的なタイミングの場合1つのスレッドが1つのノードと通信するのに要する時間は(1)式の T_1 と同じであり、取り得る最大のスレッド数 $N_{S_{max}}$ は

$$N_{S_{max}} = \frac{tc_1}{ts_1 + ts_2 + td_1 + td_2} + 1 \quad (3)$$

である。

スレッド数を N_s とし、 $N_s \leq N_{S_{max}}$ のとき1回の更新周期内に1つのスレッドが通信できる最大のノード数は p/T_1 なので、システム全体として通信できるノード数 n_p は

$$n_p = N_s \frac{p}{T_1} \quad (4)$$

である。ネットワーク上のノードが単位時間内にサーバに渡せるデータの総量、トータルスループット (T_{TP}) は各ノードがサーバに渡す1回のデータ量を L_d とすると、 $N_s \leq N_{S_{max}}$ の範囲において

$$T_{TP} = \frac{n_p L_d}{p} = \frac{N_s L_d}{T_1} \quad (5)$$

であり、 $N_s > N_{S_{max}}$ の領域においては

$$T_{TP} \leq \frac{N_{S_{max}} L_d}{T_1} \quad (6)$$

が予想される。

必要最小のスレッド数 $N_{S_{min}}$ は n_p が $n_p \geq m$ を満たす最小の N_s である。したがって、(6)式より

$$N_{S_{min}} = \min \left\{ N_s : N_s \geq \frac{m T_1}{p} \right\} \quad (7)$$

となる。

また、(4)式よりスレッド数 $N_{S_{min}}$ のとき、1つのスレッドが時間 p 内に通信できるノード数 $n_{p_{max}}$ は

$$n_{p_{max}} = \max \left\{ \text{int}(n_p) : \text{int}(n_p) \leq \frac{p}{T_1} \right\} \quad (8)$$

である。

従って、全てのノードと通信するための最適なスレッド数 $N_{S_{opt}}$ は次の式で求められる。

$$\begin{aligned} N_{S_{opt}} &= \min \{ \text{int}(N_s) : \text{int}(N_s) n_{p_{max}} \geq m \} \\ &= \min \left\{ \text{int}(N_s) : \text{int}(N_s) \geq \frac{m}{n_{p_{max}}} \right\} \end{aligned} \quad (9)$$

ここで、もともと $N_{S_{opt}}$ が(3)式で求めた $N_{S_{opt}} \leq N_{S_{max}}$ を満たさない場合このシステムは要求された周期内でのデータの更新は出来ないので、通信レートをさらに高速のインターフェースを使うか、アプリケーションソフトウェアの $ts_1 + tc_1 + ts_2$ を小さくする手立てが必要である。また、高速の通信においてはソフトウェアの応答時間に比べ回線上にデータが流れている時間は比較的小さく、システムのスループットがソフトウェアによって大きく左右される場合が多い。

3.6 通信性能の評価

ここでは実際の風車遠隔監視システムへの適用例を示し、スループットを評価する。

イーサネットのプロトコルの規格 IEEE802.3 では OSI モデルの物理層とデータリンク層のみを規定している。しかしながら、一般のコンピュータが備えている通信ドライバソフトウェアはもっと上位の層、トランスポート層およびネットワーク層まで規定した、TCP/IP や UDP/IP プロトコルに対応しているものが殆どである。従って、汎用性を損なわずに柔軟にパケットを構築できるようにするため本システムでは UDP/IP を採用した。

検討したシステムにおいてノード数は100、通信ビットレートは10Mbps で全ノードのデータ更新周期は1秒である。また、風車制御装置のソフトウェアの応答時間 tc_1 は最大150m秒、設計上の ts_1 、 ts_2 はそれぞれ、最大2m秒、3m秒である。

ここで、サーバから風車制御装置に送信するパケットは風車の監視データの要求パケットで、風車制御装置からサーバに送信するパケットはそれに対する応答としての監視データのパケットである。要求パケットのサイズは95バイト、監視データのサイズは1397バイトなので、10Mbps で通信した場合、図4において td_1 は76 μ S、 td_2 は1.117mS である。(3)式より $N_{S_{max}} =$

24.22なので N_s は $N_s \leq N_{s_{max}} = 24$ でなければならない。次に、(7)式、(8)式から必要スレッド数 $N_{s_{min}}$ 及び1秒以内に通信できる最大ノード数 $n_{p_{max}}$ を求めると、 $N_{s_{min}} = 12$ 、 $n_{p_{max}} = 6$ である。従って、(9)式より最適なスレッド数 $N_{s_{opt}}$ は $N_{s_{opt}} = 17$ であり、 $N_{s_{opt}} \leq N_{s_{max}} = 24$ を満たしている。

次に、実際にサーバとノード間の通信を行って、スレッド数を変えながら、スループットを測定し計算値と測定値の傾向を比較した結果を、図5に示す。ただし、ここでのノード数は測定のため用意できたコンピュータの台数の都合から10であり、各ノードとの通信データサイズは $1397 + 95 = 1492$ バイトである。また、図において○は測定値であり、実線の増加部分は式(5)、 $N_{s_{max}} = 24$ を超えた一定値の部分は式(6)の条件より求めた計算値である。

図5に示すとおり、実測したスループットは計算値にほぼ一致しており、計算式の妥当性が確認できた。また、スレッド数が10の時、平均のトータルスループットとして、約800kbpsを得ることが出来ており、スレッド数を17に増やして最適に設定することで、更に多数のノードと接続した場合でもシステムとして必要なスループットを得られる可能性を示した。

図5において、スレッド数が5以下の領域で実測値が計算値に比べ高いスループットを示しているのは、実際の tc_1 の平均値が計算で使用した値（最大値）より小さかったため、システムとしてマージンが増える方向でありマルチスレッドの設計上問題とならない。

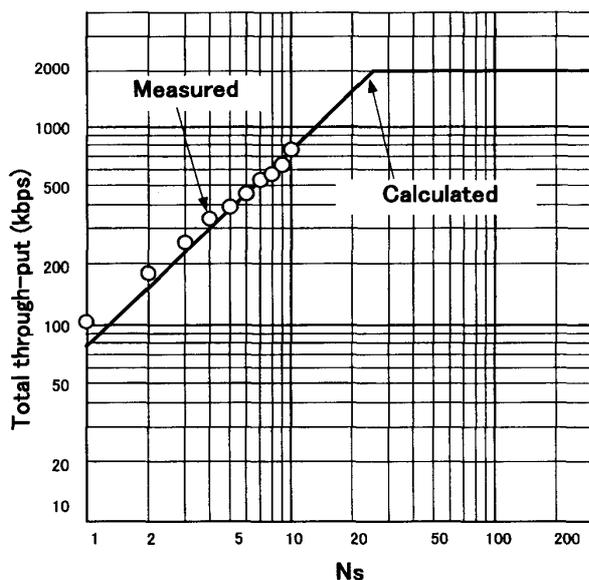


Fig. 5 Relationship between N_s and total throughput.

4. まとめ

本論文では、風力発電設備（ウィンドファーム）用遠隔監視システムの構成及び機能について述べ、大規模風車サイトにおけるローカルネットワークに関し以下の結論を得た。

- (1) サイトのローカルネットワークにはイーサネットが十分適用できる。
- (2) 本システムにおけるスレッド数とネットワークのスループットの関係を理論的、実験的に明らかにした。また、定められた周期内でのオーバーヘッドを小さくするためにはマルチスレッドが効果的であることを示した。
- (3) データの更新周期、通信データサイズ、通信レート及びアプリケーションソフトウェアのオーバーヘッドに基づく最適なスレッド数の設計手順を示した。また、その方法で求めた最適なスレッド数は100台の風車において17である。スレッド数を17とした場合のネットワークのトータルスループットは1.12Mbps以上が予想され要求されるスループットを満足することを理論的に明らかにした。
- (4) 実験により、(3)の設計手順の妥当性を確認した。また、スレッド数が10のとき、平均のトータルスループットとして、約800kbpsを得ることが出来ており、スレッド数を17に増やして最適に設定することで、更に多数のノードと接続した場合でもシステムとして必要なスループットが得られる可能性を示した。

本論文では実測試験において、機材の関係で実際にはノード数が10とネットワーク負荷としては比較的軽い領域での試験となった。そのため、ネットワーク上でのデータの衝突の影響やスレッドが $N_{s_{max}}$ を超えた時のスループットについて確認できなかった。今後、接続するノード数を増やして、スループットの限界について検討する予定である。

また、本論文では、新設する風力発電設備の遠隔監視システムにおけるローカルネットワークについて検討したが、一方、過去に建設されて稼働中の通信機能の貧弱な設備については十分な遠隔監視が出来ておらず、そうした古い設備が風力発電事業者の運転保守におけるコストや作業環境の負担になっている。遠隔監視システムを、旧設備に導入するためのインターフェースの検討も今後の課題である。

最後に、通信の性能試験をするにあたりノード側ソフトウェアを準備してくれた長菱制御システム株式会社の増田隆幸君に謝意を表す。

参 考 文 献

- 1) 高橋政司：“エネルギーの現状と展望及び新エネルギー関連施策の概要”，NEDO，新エネルギー導入実践講座，pp.5-36，1999.
- 2) 窪谷達雄：“風力エネルギー”，火力原子力発電技術協会，火力原子力発電，Vol.49，No.10，pp.48-54，1998.
- 3) 山田俊郎，猪股登：“風力発電システムの技術動向と設計事例”，電気設備学会誌，Vol.20，No.2，pp.67-70，2000.
- 4) 江藤春日，深堀潔，福田光芳：“WEB技術を用いた風車遠隔監視システム”，菱制技報，Vol.4，pp.20-25，2001.
- 5) Adam Denning：“Visual C++ Inside Out”，Microsoft Press, (株)アスキー，1997.
- 6) 江藤春日，日高健吾，宮川健，松永誠：“風車用コントローラ（MRC）の開発”，菱制技報，Vol.4，pp.7-13，2001.
- 7) Adam Denning：“ActiveX controls Inside Out”，Microsoft Press, (株)アスキー，1997.