STUDIA INFORMATICA 2008
Volume 29 Number 4B (81)

Arkadiusz JESTRATJEW, Karol OPIELKA, Jan ZIENTEK
Silesian University of Technology, Institute of Informatics
Rafał MAKOWSKI
Mining Electronics Factory "ZEG" Inc.

DISTRIBUTED CONTROL SYSTEM FOR EXPLOSION HAZARDOUS AREAS

Summary. The paper covers an architecture of a distributed SCADA system built with Microsoft .NET Framework for extremely low bandwidth field networks used in explosion-hazardous areas. A control system for mining conveyor belts, developed by Mining Electronics Factory "ZEG" Inc., is used as an example of such field network. Some performance issues are also considered.

Keywords: SCADA, explosion hazardous area, .NET Framework

ROZPROSZONY SYSTEM STEROWANIA URZĄDZENIAMI PRACUJĄCYMI W STREFACH ZAGROŻENIA WYBUCHEM

Streszczenie. W artykule przedstawiono architekturę rozproszonego systemu kontrolno-nadzorczego SCADA, opartego na platformie Microsoft .NET Framework. System ten umożliwia sterowanie i akwizycję danych za pośrednictwem sieci polowej o bardzo małej przepustowości, wykorzystywanej w strefach zagrożenia wybuchem. Jako przykład wykorzystano system sterowania górniczymi przenośnikami taśmowymi opracowany przez Zakład Elektroniki Górniczej SA. Pod uwagę wzięto także zagadnienia wydajności aplikacji.

Słowa kluczowe: SCADA, strefa zagrożenia wybuchem, .NET Framework

1. Introduction

Explosion-hazardous areas exist in many industries, namely mining, petrol, ships and others. Staying inside such areas is always a threat of personal injury or even death. Reduction of a time span that a person stays inside is therefore an important design principle. One

may achieve that principle with extensive usage of automatic field devices and remote data acquisition, visualization and control systems. A mining conveyor belts control system named USPP-05, that is developed by Mining Electronics Factory "ZEG" Inc. [1, 2, 6] is used across this paper as an example of such control system.

Any use of electricity in explosion-hazardous areas is subjected to several restrictions. Design of modern computer control systems used in such areas is also affected by these restrictions, particularly when it comes to communication. In order to achieve safety principles, the intensity of current passed through copper transmission lines must be strictly reduced. That leads to extremely low baud rates available for data exchange, often less than a few kilobits per second. Because of low power and noisy environment, there is also relatively high rate of transmission errors. With fiber-optic transmission lines one can overcome these limits, however there are also some drawbacks, mainly in terms of reliability and serviceability in some environments. In addition, modems, barriers and repeaters used for long distance transmission have their own latency times, further reducing possible data exchange rates.

Distributed SCADA (*Supervisory Control And Data Acquisition*) systems that cooperate with such low speed field networks needs to be explicitly designed to take full advantage of the available transmission bandwidth. In case that multiple clients exist, some form of cache memory ought to be maintained. The cache memory allow to accelerate completion of at least some requests made by these clients. These requirements leads to a three-layer architecture shown in Fig. 1.

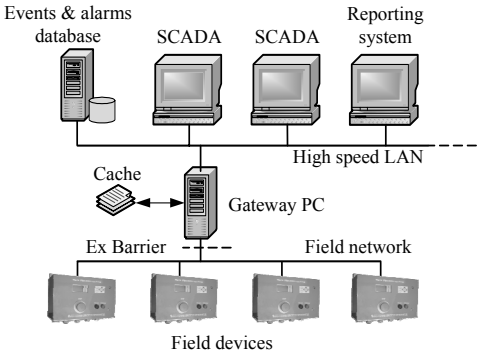

Fig. 1. System architecture overview
Rys. 1. Ogólna architektura systemu

The distributed control system is built of a set of controllers located in hazardous area (in the example – mine drift). These controllers are connected by means of specialized computer network (a field network) and exchange data using modems and repeaters as/if necessary. Parts of the system are isolated from each other with barriers.

Outside the hazardous area there is a standard PC computer that provide a bridge between field devices located in explosive area, available over extremely low speed field network, and multiple visualization, engineering and reporting stations connected to high speed corporate

LAN network. This dedicated PC runs a protocol converter software that ensures security precautions, translates user commands into proprietary field network protocol frames, retrieves device responses and provides high level data interface available for multiple LAN clients using standard network protocols. The protocol converter software also manages a data cache in order to speed-up data access for multiple clients requesting the same data.

2. Safety and security considerations

Safety is an important aspect of the system architecture. Standard PC computers and software cannot be considered reliable. Long running field network communication cables, located in hostile environment are also subject to damage. These events may lead either to loss of data acquisition ability or loss of control over field devices. To prevent negative consequences of such failures, all safety requirements have to be fulfilled by field devices located in hazardous area, so gateway PC hardware, software or communication cable failures can be tolerated. Development of field devices shall also be done in accordance with specific laws. To avoid any data loss during failure, field devices ought to provide some data buffering capabilities.

Remote commands capability may also threaten system safety. There may exist commands that are inconsistent in some way with system state and thus must not be ordered. For example, a “start conveyor belt” remote command must not be ordered when the conveyor is under maintenance. Safeguards must me implemented in field devices to ignore inappropriate remote commands, even if ordered by authorized persons.

Remote control over field devices ought to be secured. Even if a command is entirely safe (for example “stop conveyor belt” command) it may disturb normal process workflow, generate unnecessary downtime etc. Thus command ordering shall be restricted to authorized users with appropriate permissions held.

From security point of view, the gateway PC software provides secured entry point into field network and field devices. The gateway software performs authorization of remote users’ requests based on application roles.

Proper design of security-related software is not a trivial task, and thus extensive use of operating system support for user authentication and management is crucial to avoid possible drawbacks. More information on security-related software design and implementation may be found in [3].

The high-speed LAN is divided into two distinct parts – an engineering network and a corporate network – Fig. 2. These parts are connected through a firewall that protects engineering network clients and the gateway PC itself from typical security threats that may exist

in the corporate network, like Internet worms, viruses or unauthorized data access requests. The firewall may also restricts engineering network stations access to the Internet.

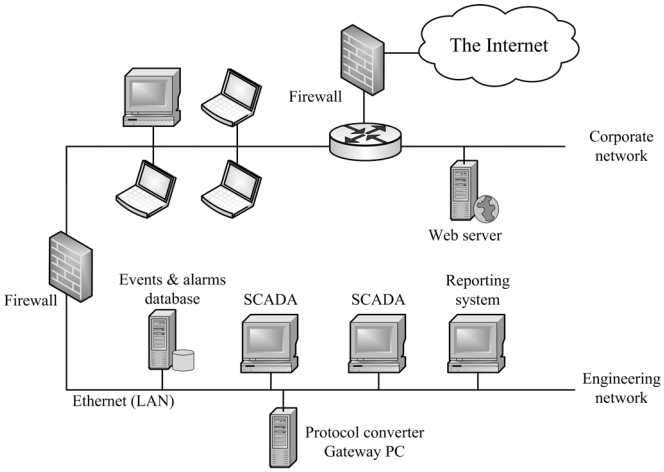

Fig. 2. Engineering network and corporate LAN interconnection
Rys. 2. Połączenie sieci nadzorczej i korporacyjnej sieci LAN

The engineering network is considered to be safe and trusted, clients that reside there are provided with full service, however clients located in the corporate network have limited trust and are able to perform monitoring and reporting activities.

With support of the corporate infrastructure – Web server, firewalls, etc. – it is relatively easy to make these services available in the Internet for authorized users, however appropriate security precautions must be taken.

3. System software architecture

A logical view of the software architecture is shown in Fig. 3. Components outlined with dotted lines have not been implemented yet. Protocol converter core components are colored grey. Horizontal slashed line depicts layer boundaries and communication interfaces used to cross them.

The software is built with Microsoft .NET Framework 2.0 and based on "software PLC" concept introduced in [4]. Similar architecture has already proved its value in mine pump station control system.

The main advantages of using .NET Framework include:

- increased code reliability compared to native C++ programming,
- configurable inter-process communication with .NET Remoting – fast with binary data serialization over TCP/IP or standard compatible and easy to consume with XML data serialization over HTTP,
- rich base class library to speed up software development,

- acceptable costs of use in terms of CPU ticks and memory usage [4].

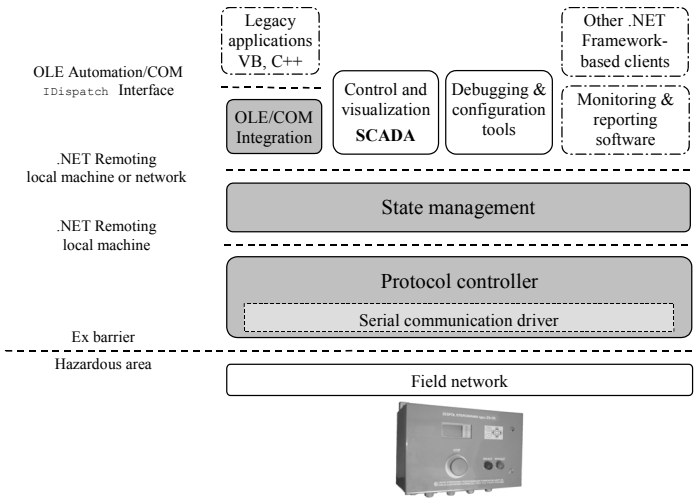

Fig. 3. System software components
Rys. 3. Komponenty oprogramowania systemu

Note that both state management and protocol controller components are designed to run on a single PC computer, however these components are hosted in separate processes, and thus communicate using .NET Remoting. Client software: SCADA systems, reporting services & database etc. ought to be hosted outside protocol converter PC to save its valuable resources. However, a single instance of SCADA application running on protocol converter PC increases reliability of the whole system and thus in many cases is worth the resources it consumes.

3.1. State management

The component acts as a proxy between Protocol controller component and multiple clients that use the system. It provides a well defined and stable interface to retrieve system state information and synchronizes multiple clients calls.

The component uses a paradigm of shared memory that is divided into number of zones. The concept is similar to the memory organization found in programmable logic controllers (PLCs). Each zone is an array of integers and is assigned individual purpose and size. The strict distinction exists between zones that are writable for individual components. Several zones are considered command zones and may be written by higher-level components (SCADA, monitoring software), the other zones are considered result zones and may be written by Protocol controller component only.

Result zones also play a role of cache memory, that is filled with data received from the slow field network, in order to achieve better response times for LAN clients. Each data item has a timestamp that provides “freshness” information to the user. State management compo-

nent is able to store its data into external durable data store during shutdown, allowing cached data surviving system shutdown and restart, increasing total data availability.

State management component is accessible using any transport protocol supported by .NET Remoting infrastructure. Remoting configuration is made with standard configuration files. Clients authentication and authorization is left to the underlying communication framework. Security configuration depends on communication protocol used and must be done by network administrator.

3.2. Protocol controller

The component handles communication protocol of the field network. It translates high-level commands issued by SCADA system or other clients into protocol-dependent frame series [2], schedules the frames and transmits them to field devices. It is also responsible for received frames analysis, error detection and retransmissions in case of error.

After receiving a response, the component applies timestamp and makes data available by storing it into appropriate memory zone managed by System state management component.

A very slow field network makes it especially important to utilize its throughput to maximum. To satisfy this requirement the protocol controller encloses a read-ahead cache manager, that issues “read state” commands to field devices in spare time. These commands are issued periodically to each field device, thus updating cached state of devices, improving field network usage. Paper [5] provides more detailed description of the two above components.

3.3. Control and visualization application

The control and visualization application provides a graphical user interface for underlying components. The application is built with Microsoft .NET Framework 2.0 and communicates with the other components using standard .NET Remoting mechanisms, either on single machine or over the network.

The application provides continuously updated status information about the whole system of conveyors in intuitive form using icons and colors. The main screen (Fig. 4) provides system overview information that is used most of the time. The status data is retrieved from the cache memory managed by the state management component, so there is no additional overhead when multiple visualization stations are used to display system status.

When more detailed information about a conveyor is required, it is likely not available in the cache and thus must be retrieved directly from field devices. That data retrieval takes out field network time, so multiple clients requesting detailed data about varying conveyors or subsystems will have definitely negative impact on status data refresh rate. However, after

the detailed data is retrieved and stored into cache memory, it is available for other clients with no additional costs and is updated on user's request only. An example conveyor details screen is shown on Fig. 5. Note individual timestamps shown for each detail in the bottom line.

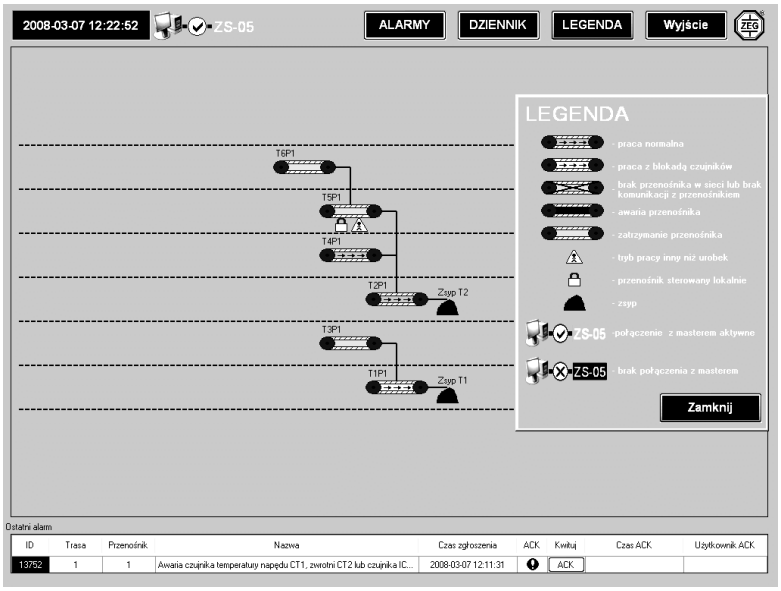

Fig. 4. Conveyor system status overview screen
Rys. 4. Widok statusu systemu przenośników taśmowych

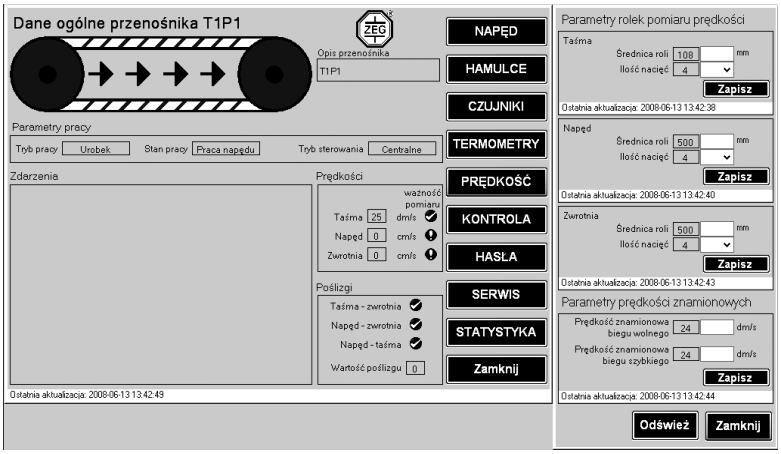

Fig. 5. Example conveyor details screen
Rys. 5. Przykładowy ekran szczegółowych informacji

Negative impact of detailed requests made by multiple clients may be mitigated if there is only a single privileged client that is allowed to make these requests and all other clients are functionally restricted to display main system status information. That is often the case in mining industry, where a single privileged controller (*a setter*) is associated to each work shift. That person is allowed to order commands and is interested in detailed conveyor data. The other clients (management, reporting etc.) mainly monitor general system working conditions.

4. Resources utilization

Total response time as seen by the user is about a few seconds and is mainly affected by transfer rate of the underlying field network and response time of field devices. Client requests rate has much less impact on total response times, however client processor utilization is noticeably affected if request rate is too high. The processor utilization vs. client request rate is depicted on Fig. 6. For high client request rate, most of the client processor time is spent on remoting calls and screen repaints – Fig. 6b. As shown on Fig. 6a, processor utilization of state management component that serves client request is relatively stable.

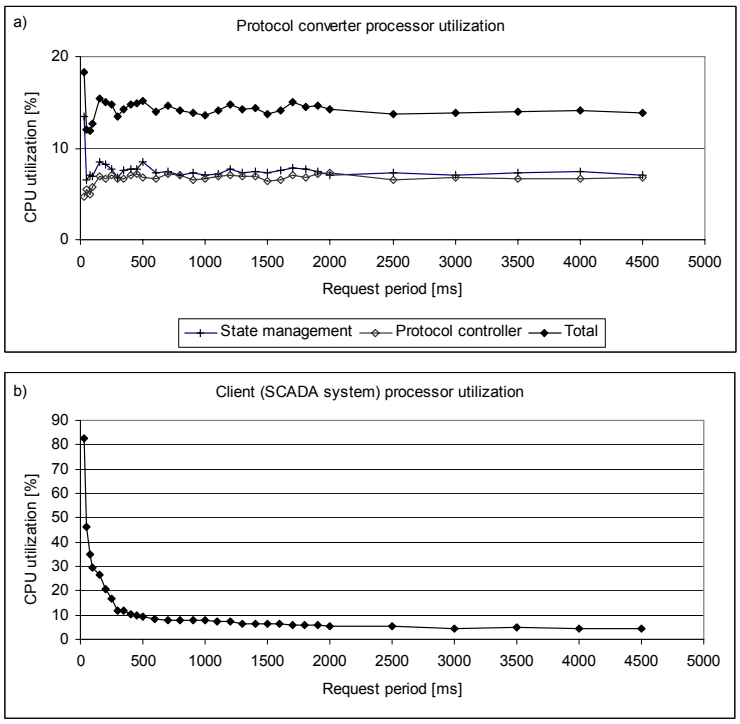

Fig. 6. Average processor utilization vs. client request rate
Rys. 6. Średnie obciążenie procesora w funkcji częstości żądań

5. Conclusions

Electronic equipment used in explosion-hazardous areas must be specifically designed to assure safety requirements. For distributed control systems it means the system must perform well with very slow communication links between system nodes. That impose special requirements on software architecture of the system in order to make data available for multiple clients. Such architecture, based on Microsoft .NET Framework, is proposed in the paper. Proposed architecture has several advantages, where built-in support for various communica-

tion protocols on the LAN side and decoupling clients from field network protocol internals seems to be the most valuable.

Proposed system architecture succeed during development of SCADA system for mining belt conveyors.

REFERENCES

1. Belt conveyor control system USPP-05. Control device ZS-05 reference manual. R418863. ZEG SA, Tychy 2007.
2. Belt conveyor control device ZS-05. Communication protocol. Internal documentation. ZEG SA, Tychy 2007.
3. Howard M., LeBlanc D.: Writing Secure Code, 2nd edition. Microsoft Press, 2002.
4. Jestratjew A., Gaj P.: Bezpośrednie sterowanie procesem technologicznym w środowisku Microsoft .NET Framework. In: Systemy informatyczne z ograniczeniami czasowymi (in Polish), WKŁ, Warszawa 2006, p. 391÷404.
5. Jestratjew A., Opielka K., Zientek J., Makowski R.: Protocol converter with cache for field networks in explosion hazardous areas. In: Węgrzyn S., Czachórski T., Kwiecień A. (eds.): Contemporary Aspects of Computer Networks, Vol. 2, WKŁ, Warszawa 2008, p. 241÷248.
6. Opielka K., Zientek J., Jestratjew A., Domagała W., Makowski R., Nowakowski A., Bywalec A.: Supervisory Control System for Mining Belt Conveyors. To appear in: Proc. of 17th Intern. Conf. on Automation in Mining ICAMC'2008, Kraków, 7-11 Sept. 2008.

Recenzent: Dr hab. Tadeusz Wieczorek, prof. Pol. Śląskiej

Wpłynęło do Redakcji 26 września 2008 r.

Omówienie

Rozproszone informatyczne systemy sterowania i wizualizacji są stosowane w celu ograniczenia czasu przebywania ludzi w strefach zagrożenia wybuchem, jakie występują w wielu branżach przemysłu, np. w kopalniach, przemyśle stoczniowym czy petrochemicznym. Urządzenia sterujące i okablowanie pracujące w takich strefach muszą spełnić wiele wymagań związanych z bezpieczeństwem i niezawodnością ich funkcjonowania. Spełnienie tych wy-

magań w przypadku rozproszonych systemów sterowania prowadzi do znacznego ograniczenia przepustowości polowej sieci komunikacyjnej łączącej węzły systemu rozproszonego.

Budowa rozproszonego systemu wizualizacji i kontroli SCADA wymaga połączenia powolnej sieci polowej oraz szybkiej sieci LAN przedsiębiorstwa (rys. 1). Elementem pośredniczącym jest komputer PC z oprogramowaniem konwertera protokołów, pozwalającym udostępnić dane z sieci polowej wielu klientom, takim jak system wizualizacji czy raportowania, za pośrednictwem standardowych mechanizmów jak TCP/IP i XML.

Ze względu na małą niezawodność komputerów PC oraz niebezpieczeństwo uszkodzenia okablowania komunikacyjnego wszystkie funkcje zabezpieczeń muszą być realizowane lokalnie przez urządzenia sterujące w strefie zagrożenia wybuchem. Wydawanie poleceń sterujących pracą urządzeń musi być ograniczone do nielicznych uprawnionych użytkowników. W tym celu można wydzielić z sieci LAN przedsiębiorstwa zaufaną część (rys. 2) i ograniczyć funkcjonalność dostępną dla pozostałych abonentów do monitorowania procesu.

Oprogramowanie systemu (rys. 3) zrealizowano z użyciem Microsoft .NET Framework, jako mechanizm komunikacji wykorzystano .NET Remoting. Pozwoliło to uzyskać dużą elastyczność i pewność działania, przy akceptowalnych narzutach [4]. Widok aplikacji SCADA przedstawiono na rys. 4 i 5. Dzięki buforowaniu informacji uzyskanych z powolnej sieci polowej możliwa jest jednoczesna praca wielu klientów. Na rys. 5 widoczne są znaczniki czasu odebrania informacji z sieci polowej. Na rys. 6 pokazano obciążenie procesora w funkcji częstości odświeżania wizualizacji. Czas odpowiedzi systemu na żądania użytkowników jest zależny głównie od czasu transmisji informacji w sieci polowej, który jest rzędu sekund.

Addresses

Arkadiusz JESTRATJEW: Silesian University of Technology, Institute of Informatics, ul. Akademicka 16, 44-100 Gliwice, Poland, arkadiusz.jestratjew@polsl.pl

Karol OPIELKA: Silesian University of Technology, Institute of Informatics, ul. Akademicka 16, 44-100 Gliwice, Poland

Jan ZIENTEK: Silesian University of Technology, Institute of Informatics, ul. Akademicka 16, 44-100 Gliwice, Poland

Rafał MAKOWSKI: Mining Electronics Factory "ZEG" Inc., ul. Burshego 3, 43-100 Tychy, Poland