

Ralph Langner

Ethernet ausgetrickst

Bei der direkten Kopplung der Siemens-Steuerungen S5 und S7 an das Office-Ethernet weist die Protokoll-Unterstützung deutliche Schwächen hinsichtlich Datensicherheit und Netzauslastung auf. Ein einfaches Verfahren beseitigt dieses Manko: mittels gängiger Step7-Befehle und ohne den Zukauf bestimmter Produkte.

Während im Automatisierungsumfeld noch diskutiert wird, ob Ethernet der Feldbus der Zukunft ist, ist die Entscheidung bei der Datenbrücke zu IT-Anwendungen längst gefallen: Als Medium zur Datenübertragung dominiert hier unangefochten TCP/IP.

Auch Siemens setzt bei den Ethernet-CPs 343, 443 und 1430 seit Jahren auf Ethernet-TCP/IP. Allerdings birgt die direkte Ethernet-Anbindung der S5 oder S7 bei näherem Hinsehen ein Manko: die unzureichende Protokollunterstützung. Siemens ist hier den Weg vieler anderer Hersteller gegangen, die traditionell über

Punkt-zu-Punkt-Medien verwendeten Protokolle eins zu eins auf TCP/IP umzusetzen. Dies funktioniert zwar prinzipiell, hat aber seinen Preis: Denn anders als Punkt-zu-Punkt-Medien wie die serielle RS-232- oder 20-mA-Schnittstelle ist Ethernet ein Bus, den sich alle angeschlossenen Stationen über ein konkurrierendes Zugriffsverfahren teilen – woraus letztlich auch die Echtzeit-Problematik resultiert. Die Folge: Das permanente Abfragen von Datenbausteinen bewirkt eine ständige Netzlast – und zwar auch dann, wenn sich die Daten überhaupt nicht geändert haben. In Konsequenz

sinkt die insgesamt verfügbare Bandbreite erheblich, wenn eine Vielzahl von Steuerungen am selben Netz hängt.

Als ein weiteres Problem ergibt sich die beim Polling-Verfahren nicht garantierte Konsistenz und Lückenlosigkeit von Datensätzen. Denn während sich der Datenaustausch für eine Prozessvisualisierung mehr oder weniger kontinuierlich darstellt, sieht das für andere Anwendungen wie die Betriebsdaten- und Maschinendatenerfassung (BDE/MDE) ganz anders aus: Hier werden buchungspflichtige Datensätze übertragen, die nur zu einem bestimmten Zeitpunkt entstehen – zum

(Bild: Computer & AUTOMATION, Quelle: Langner/Siemens)

Beispiel beim Los- oder Schichtwechsel – und anschließend „genullt“ werden, also nicht mehr existent sind. Ergo muss die IT-Anwendung beim Polling-Verfahren exakt zum richtigen Zeitpunkt lesen. Schwierig ist dies deshalb, da die Steuerung den korrekten Zeitpunkt in der Regel kennt, der PC jedoch nicht. Der PC kann zwar versuchen, sehr häufig zu lesen, um „den“ gültigen Datensatz mitzubekommen – dies verursacht aber gerade für Daten, die sich nur wenige Male pro Schicht ändern, eine übertrieben hohe Netzwerklast.

Abfragezyklen eingespart

Wie lässt sich dieses Dilemma beheben?

Die Siemens-Lösung für das Problem heißt „Send/Receive-Schnittstelle“. Diese Schnittstelle ermöglicht es dem Automatisierungstechniker, Daten aktiv zum PC zu senden. So spart sich der PC die permanenten Abfragezyklen, die in 99 % der Fälle nur redundante Daten liefern und lediglich die Netzlast erhöhen. Konkret funktioniert das folgendermaßen: Ein Aufruf der Funktionsbausteine AG_SEND – bei den Steuerungen S7-300 und S7-400 – und AG_LSEND – bei der S7-400 zum Versenden größerer Datenblöcke – schickt ein komplettes Datenpaket zum PC, zum Beispiel einen kompletten BDE-Buchungssatz. Der PC muss die eintreffenden Datenpakete nun lediglich entgegennehmen und korrekt verarbeiten – beispielsweise mit einer geeigneten Komponentensoftware. Hierfür gibt es verschiedene Produkte für Visual Basic, Delphi und C++ am Markt – beispielsweise LUCA von Langner Communications –, die empfangene Pakete über Callback-Funktionen an die Anwendung melden; mit der Winsock-Schnittstelle lässt sich das Ganze auch selbst implementieren. Anstatt in einer Endlos-Schleife Fetch-Befehle abzuschicken, kann der PC-Programmierer also wie gewohnt ereignisgesteuert implementieren: Werden Empfangsdaten vom Treiber signalisiert, hat die SPS ein neues Telegramm geschickt.

Zur Nutzung dieses Nachrichten-orientierten Verfahrens erweisen sich zwei Hilfsmittel als wertvoll: Das erste besteht darin, dass jedes Datenpaket schon auf der SPS ein Typenfeld erhält, das im ersten Wort des Pakets untergebracht ist. Auf diese Weise ist es möglich, unterschiedliche Datenpakete zu verwenden; zum Beispiel 1 für BDE, 2 für MDE, 3 für Prozessdaten/Drücke, 4 für Prozessdaten/Temperaturen und 5 für Qualitätsdaten.

Die zweite Option ist ein zusätzlich zum Pakettyp mitgeführter Sequenzzähler, den die SPS bei jedem Sendevorgang hochzählt. So kann der PC auf einfache Weise Paketverluste erkennen, die beispielsweise bei einer gestörten Netzwerkverbindung entstehen.

Repair-Programm inklusive

Auf Basis dieses Verfahrens ist eine ereignisgesteuerte Datenübertragung von der SPS zum PC mit gleichzeitigem Erkennen von Daten-

Die Send/Receive-Schnittstelle

Um Daten aktiv von der Steuerung zum PC zu schicken, bietet sich die Send/Receive-Schnittstelle der Steuerungen an. Dabei wird im ersten Schritt eine ISO-on-TCP-Verbindung projektiert. Im zweiten Schritt werden die zu sendenden Daten in einen Datenbaustein kopiert und im dritten

Schritt Pakete mit den Funktionsbausteinen AG_SEND oder AG_LSEND abgeschickt. Für die Gegenrichtung wird, falls erforderlich, eine Write-Verbindung verwendet. Benötigt jemand besonders hohe Performance, empfiehlt sich die Projektierung von UDP anstelle von ISO-on-TCP.



Projektion der Send-Verbindung: Für den aktiven Nachrichtenversand wird eine ISO-on-TCP-Verbindung projiziert. Der TSAP der Verbindung ist wichtig, er muss der PC-Anwendung für den Verbindungsaufbau bekannt sein. Als Betriebsart wird „Send/Recv“ eingestellt.

verlusten realisierbar. Aber mit dem Erkennen allein ist es noch nicht getan!

Mit Hilfe einer kleinen Erweiterung – einem auf der SPS angelegten Datenpuffer – lassen sich erkannte Datenverluste auch reparieren. Je nach Pakettyp und Paketlänge kann dieser Puffer ein oder mehrere Pakete zwischenspeichern. Hierfür genügt es, den entsprechenden Platz in freien Datenbausteinen zu reservieren, die anschließend als Ringpuffer verwaltet werden. Das SPS-Programm muss sich nur für jeden Pakettyp den zugehörigen Pufferspeicher „merken“ und im Rundlauf-Verfahren füllen. Ein Beispiel: Für BDE-Daten werden die Datenbausteine 8, 9, 10 und 11 als Puffer reserviert. Nach dem ersten Sendeaufruf wird das gesendete Telegramm – inklusive Typenfeld und Sequenzzähler – in den Datenbaustein 8 kopiert. Nach dem zweiten Sendevorgang ist Datenbaustein 9 das Ziel des Kopiervorgangs, dann Datenbaustein 10, anschließend Datenbaustein 11. Danach geht es wieder bei Datenbaustein 8 los.

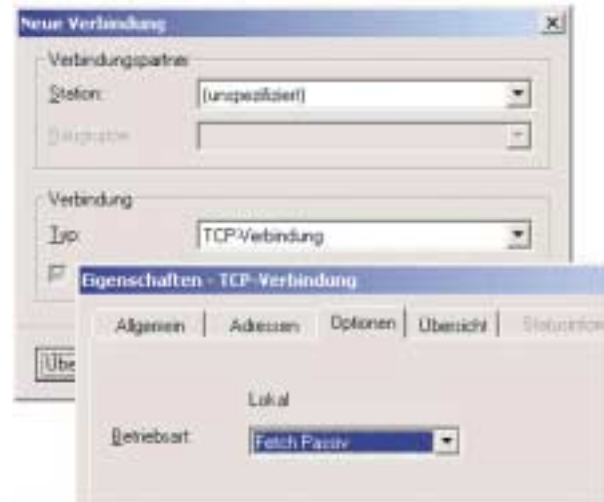
Durch dieses Vorgehen lässt sich für buchungspflichtige Datensätze eine sehr hohe Ausfallsicherheit erreichen. Da sich diese Datensätze zudem nur selten ändern, reichen wenige Puffer auf der Steuerung aus, um Netzwerkstörungen über viele Stunden hinweg zu überbrücken. So reichen beispielsweise im Zweischicht-

betrieb bereits vier Datenbausteine aus, um die aufsummierten Maschinendaten von zwei Tagen zu sichern.

Doch wie gelangen die Daten bei wiederhergestellter Netzwerkverbindung aus dem Ringpuffer zum PC?

Fetch-Verbindung für den CP

Eine – relativ aufwendige – Methode wäre, eine Logik auf der SPS zu implementieren, die zunächst den Puffer abarbeitet, die dort vorgefundenen Daten verschickt und erst im Anschluss daran weitere „frische“ Datenpakete sendet. Wesentlich einfacher zu implementieren und praxisorientierter ist allerdings eine andere Methode: Hierbei wird für die Abarbeitung des Puffers eine Fetch-Verbindung für den CP projiziert, über die der PC bei einem anhand der Sequenznummern erkannten Paketverlust selbsttätig den Ringpuffer der SPS „durchforstet“ und fehlende Pakete restauriert. Ein Beispiel: Das PC-Programm erhält über die Send-Schnittstelle die BDE-Datensätze Nummer 347 und 349, 348 fehlt. Über die Fetch-Verbindung wird nun der erste Pufferspeicher (Datenbaustein 8) gelesen. Ist dort nicht Pakettyp = 1 und Sequenznummer = 348, liest das PC-Programm die weiteren Datenbausteine 9, 10 und 11, bis der betreffende Datensatz gefunden oder alle Puffer er-



Für die Fetch-Verbindung zum Einlesen des Pufferspeichers genügt TCP. Als Betriebsart wird „Fetch passiv“ eingestellt. Der PC kann nun zu dem bei „Adressen“ eingestellten TCP-Port eine Fetch-Verbindung aufbauen.

folglos gelesen wurden (Pufferüberlauf). Eine solche Kombination aus Send/Receive und Fetch/Write ermöglicht auf Grund der stark reduzierten Netzlast auch den Aufbau großer Produktionsnetzwerke mit mehreren hundert Steuerungen.

Die Implementierung selbst ist kein Zauberwerk: Auf der SPS-Seite kann der Automatisierungstechniker die Logik mit kostenlosen „Bordmitteln“ – Step7-Befehlen und CP-Funktionsbausteinen – selbst implementieren. Auf der PC-Seite lassen sich handelsübliche Komponenten-Bibliotheken dazu nutzen, das Gegenstück in Visual Basic oder anderen Programmiersprachen zu implementieren; wer kein Geld ausgeben will, kann alles „zu Fuß“ mit Winsock implementieren. Erste Standard-Software-Produkte gibt es für dieses Verfahren auch bereits: Langner Communications bietet beispielsweise unter der Bezeichnung „Industrial Ethernet Powerpack“ entsprechende Step7-Funktionsbausteine mit passendem OPC-Server an. *im*

Nähere Informationen:
www.langner.com



Ralph Langner

ist Vorstandsvorsitzender von Langner Communications, Hamburg.

(Grafiken: Langner)