SIEMENS

SIMATIC NET

PC 软件
PG/PC 工业通信第 1 卷 - 基础知识

系统手册

11/2022

C79000-G8952-C172-16

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

⚠危险
表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。

⚠警告
表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。

⚠小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意
表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。 由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

⚠警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 ® 的都是 Siemens AG 的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
德国

文档订购号: C79000-G8952-C172
Ⓟ 10/2022 本公司保留更改的权利

目录

前言

1

1.1 欢迎使用 SIMATIC NET

SIMATIC NET - 开创书面形式的成功解决方案之先河

既然您已做出决定，我们就会支持您。 在成功应用 SIMATIC NET 的道路上，本文档将是您忠实的伴侣。 它将为您提供简单明了的主题介绍，并向您展示如何安装和组态各个组件以及如何基于 OPC 创建您自己的程序。 您将看到 SIMATIC NET 工业通信将为您、您的自动化解决方案带来契机；最重要的是，将为您的公司开启成功之门。

SIMATIC NET - 正确的决定

您了解了分布式自动化系统的优势并希望优化使用工业通信。 您期待与实力雄厚的合作伙伴合作并生产出可靠的创新产品。 SIMATIC NET 是您正确的选择。

本文档将积累您的知识并让您从专家的专有技术和专业知识中获益。

您是初学者？

利用本手册可系统性地熟悉相关内容。 请从介绍工业通信的这**第 1 卷**开始。 在此，您将找到关于通信原理和 SIMATIC NET OPC 服务器功能范围的所有必要信息。 阅读 OPC 接口的基本知识，熟悉协议及其优势和功能。

您是专业人员？

利用本手册可立即开始工作。 **第 2 卷**为您提供使用 SIMATIC NET 所需的全部信息。

第 2 卷 – 接口，条目 ID：

61630140 (http://support.automation.siemens.com/WW/view/zh/61630140)

发现示例很实用？

可灵活利用所提供的示例程序将您自己的想法付诸实践。

商标

下文的一些名称以及可能的其它名称不带注册商标符号 ®，它们均为 Siemens AG 的注册商标：

HARDNET, SOFTNET, CP 5612, CP 5613, CP 5614, CP 5622

Industry Online Support

除产品文档外，以下 Internet 地址还提供 Siemens Industry Online Support 的丰富全面的在线信息平台：
链接：(https://support.industry.siemens.com/cs/de/en/)

除新闻外，您还可以在其中找到以下内容：

- 项目信息：手册、常见问题解答、下载资料、应用程序示例等
- 联系人，技术论坛
- 提交支持查询的选项：
  链接：(https://support.industry.siemens.com/My/cn/zh/requests)
- 我们的服务提供：

  针对我们的产品和系统，我们还提供大量服务，支持机器或系统使用的每个阶段 - 从规划和实施到调试，直至维护和现代化。

有关联系数据，请访问以下 Internet 网址：
链接：(http://www.automation.siemens.com/aspa_app/?ci=yes&lang=zh)

SITRAIN - Training for Industry

该培训包括 300 多门与基本主题、扩展知识和专业知识相关的课程，以及个别部门的高级培训 - 可在 130 余个地点开展培训。课程也可单独组织，并于您的所在地进行授课。

有关培训课程以及如何联系客户顾问的详细信息，请访问以下 Internet 网址：

(https://new.siemens.com/global/en/products/services/industry/sitrain/personal.html)

安全性信息

Siemens 为其产品及解决方案提供了工业信息安全功能，以支持工厂、系统、机器和网络的安全运行。

为了防止工厂、系统、机器和网络受到网络攻击，需要实施并持续保有全面、先进的工业信息安全概念。Siemens 的产品和解决方案构成此类概念的其中一个要素。

客户负责防止其工厂、系统、机器和网络受到未经授权的访问。只有在必要时并采取适当安全措施（防火墙和/或网络分段）的情况下，才能将这些系统、机器和组件连接到企业网络或 Internet。

有关工业安全领域中保护措施的更多信息，请访问：
(https://www.siemens.com/industrialsecurity)。

Siemens 不断对产品和解决方案进行开发和完善以提高安全性。Siemens 强烈建议您及时更新产品并且仅使用最新产品版本。如果使用的产品版本不再受支持，或者未能应用最新的更新程序，客户遭受网络攻击的风险会增加。

为确保您始终了解有关产品更新的信息，请订阅西门子工业安全 RSS 源，网址如下：
(https://www.siemens.com/cert)

固件/软件支持的说明

定期检查新固件/软件版本或安全更新并加以应用。新版本发布后，先前版本不再受支持，也不再进行维护。

SIMATIC NET 词汇表

SIMATIC NET 词汇表描述了本文档中可能使用的术语。

要获取完整的 SIMATIC NET 词汇表，请访问西门子工业在线支持，网址为：

50305045 (http://support.automation.siemens.com/WW/view/zh/50305045)

回收和处置

该产品的有害物质含量低，可以回收利用并且符合指令 2012/19/EU 对废弃电子电气设备 (WEEE) 的处置要求。

请勿将产品丢弃在公共场所。

如需按照环保要求回收和处置电子废弃物，请联系已获得电子废弃物处置相关认证的公司或西门子代表。

请注意不同国家的法规。

工业通信中的 SIMATIC NET

2

概述

如果您希望了解 SIMATIC NET® 工业通信中的通信原理和各种协议的功能范围，本章中的材料将会为您提供帮助。

它介绍了 PROFIBUS 和工业以太网通信网络的基础知识、告诉您如何在 SIMATIC NET 工作中为这些通信网络实施协议，并列出这些协议的优缺点。 在本章末尾，您将看到 PROFINET 技术和应用的概述以及如何在 SIMATIC NET 中实施 PROFINET。

遍览此章后，您将能够确定用于完成自动化任务的最合适工具。

SIMATIC NET 实际上意味着什么？

工业通信是现代自动化解决方案的支柱。 通信网络和相关产品允许在尽可能多样的自动化组件和设备之间进行全集成通信。

SIMATIC NET 是 Siemens 通信网络和产品整个系列
的名称。 各种网络可最大限度地满足自动化工程中的性能和应用需求。

SIMATIC NET 能提供什么？

SIMATIC NET 提供针对客户各项工业通信需求的解决方案。 SIMATIC NET 中的通信网络和产品是 Siemens 全集成自动化 (TIA) 的组件。 在此基础上，可使用高度集成的全面通信功能实施特定于分支的自动化解决方案。 无论使用的通信网络和产品如何，SIMATIC NET 都会简化自动化系统的调试过程。

就性能和功能范围而言，SIMATIC NET 的通信网络和产品能够以自动化金字塔的形式呈现。

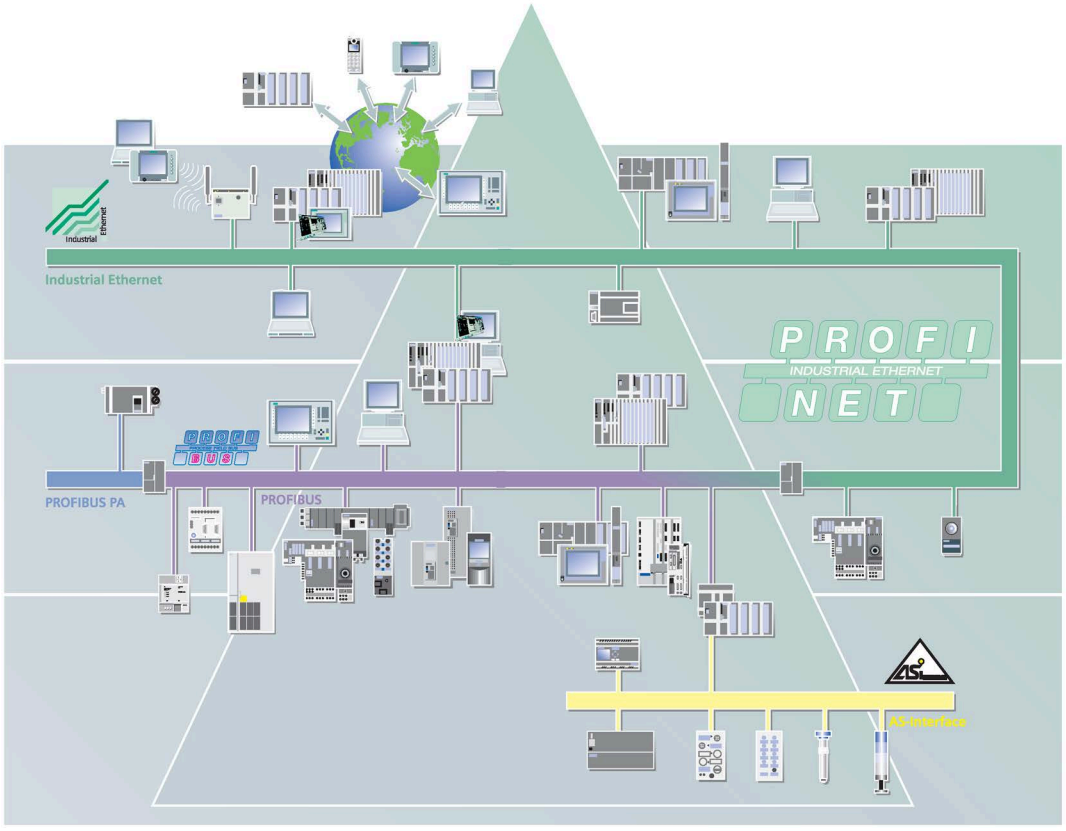

图 2-1　　SIMATIC NET 的自动化金字塔

自动化金字塔可分为三级，即现场级、单元级和管理级。

现场级是处理过程或现场通信的级别。 SIMATIC NET 为此级别提供了 PROFIBUS DP 和 AS 接口。

在单元级，采集的过程数据会分发给操作员监控的各种自动化系统或 PC。 此处，通信网络工业以太网和 PROFIBUS 在 SIMATIC NET 中使用。

更高级的管理功能在管理级处理。 在此，将对过程数据进行保存、进一步处理或将其用于分析。 对于此类任务，工业以太网适合作为通信网络。

2.1 SIMATIC NET 及协议 - 概述

SIMATIC NET 中定义了哪些协议？

SIMATIC NET 中使用了两种通信网络 - PROFIBUS 和工业以太网。 对于这两种网络，可用的协议允许在自动化组件和设备之间进行全集成通信，其功能范围可扩展，能够满足自动化工程中的各种应用要求。

下表列出的协议可与 PROFIBUS 或工业以太网结合使用：

表格 2- 1　PROFIBUS 协议

协议	说明
SEND/RECEIVE 协议	基于 PROFIBUS FDL 的简单通信服务，用来与 S5 和 S7 设备交换数据。
1 类 DP 主站	周期性读取输入数据并设置 DP 从站的输出数据。
2 类 DP 主站	周期性访问 DP 系统的诊断和调试。
DPC1	周期性读取输入数据并设置 DP 从站的输出数据，非周期性访问 DPV1 经过扩展的 DP 从站的数据记录。
DPC2	周期性访问 DP 系统的诊断和调试，非周期性访问 DP V1 经过扩展的 DP 从站的数据记录。
DP 从站	采集、转换并传输过程信号。
S7 协议	经过集成和优化的 SIMATIC S7 系统通信功能，适用于各种应用

表格 2- 2　工业以太网协议

协议	说明
SEND/RECEIVE 协议	基于传输协议的简单通信服务，用来与 S5 和 S7 设备交换数据。
S7 协议	经过集成和优化的 SIMATIC S7 系统通信功能，适用于各种应用
S7 协议（容错）	通过冗余与容错路径集成和优化的 SIMATIC S7 系统通信功能。
SNMP 协议	用于管理网络的开放协议。
PROFINET IO	PROFINET 设备间的通信

ISO/OSI 参考模型

为了更好地理解针对 PROFIBUS 和以太网所实施的协议的功能，应熟悉用于标准化各种数据通信要求的规范（即 ISO/OSI 参考模型），这一点非常重要。

“开放系统互连”(OSI) 分层模型是用于在网络中传输数据的参考模型，以国际标准化组织 ISO 的工作组命名。 它描述了 7 个按层级排列的层。 每个层都有其自己的特定任务。

作为参考模型，OSI 并非为公认标准。 但是，在电信和联网领域，许多产品均以 ISO/OSI 参考模型为导向。

图 2-2　　ISO/OSI 参考模型

ISO/OSI 参考模型中定义了哪些层？

该模型中所指定的 7 个层按三个功能级进行排列。 第一和第二层代表与硬件关联最紧密的级别、第三和第四层构成传输级，第五至第七层则实施与应用关联最紧密的级别。 各层定义如下：

- 第 1 层：
  物理层负责两个设备之间的物理连接。 它通过网络将数据从一个设备传输到另一设备。
- 第 2 层：
  数据链路层负责对数据进行可靠传输。 它将位分为多个数据块，并添加在两个设备之间传输数据所需的地址信息。 从站也负责检测链路上的错误。
- 第 3 层：
  网络层负责路由数据块并对其进行正确转发。 它处理帧的寻址及其在网络中的路由方式。 本层的一个示例为 Internet 协议 (IP)。
- 第 4 层：
  传输层协调数据包的传输。 它检查是否已完全接收所有包。 为实现此目的，需提供两个设备之间的传输连接。 第 4 层的典型示例为传输控制协议 (TCP)。
- 第 5 层：
  会话层在要于其间传输数据的两个设备之间建立更为永久的连接。 此层负责建立和终止连接以及维护连接。
- 第 6 层：
  表示层负责将数据转换为特定应用所需的格式。 该数据还准备好进行传输。 这包括数据压缩和编码。
- 第 7 层：
  应用层所提供的应用是接收数据以供进一步处理或提供数据进行传输。 此类应用的典型示例是电子邮件程序或 Internet 浏览器。

2.2 PROFIBUS 工业通信 - 概述

2.2.1 PROFIBUS - 它是什么？

PROFIBUS 定义

PROFIBUS 为开放式国际标准化 (EN50170) 总线系统，用于现场设备的过程和现场通信以及自动化单元内的数据通信。 PROFIBUS 的可应用范围从生产及过程自动化到楼宇自动化。

PROFIBUS 的主要特征有哪些？

PROFIBUS 的主要特征如下：

- 通过经济高效的通信介质（例如双绞线）进行数据传输。
- 应用广泛（得益于可编程控制器和操作员控制），并通过统一的总线监视设备通信。
- 标准化数据通信，符合 EN 50170、EC 61158（服务和协议）及 IEC 61784。

- 可在总线段中的任意点执行调试、组态和故障排除。
- 投资安全（得益于现有的 PROFIBUS 系统），可扩大投资且无不利影响。

2.2.2 PROFIBUS - 工作原理是什么？

PROFIBUS 的工作原理

PROFIBUS 规范十分灵活，允许针对不同应用领域中的特定任务实施经过优化的各种协议。 FDL 数据链路层（ISO/OSI 参考模型的第 2 层）确保能够统一控制对使用令牌传递的总线的访问。

令牌传递在 PROFIBUS 中如何工作？

令牌传递控制对总线的访问；换句话说，仅允许发送当前拥有令牌的总线节点。 固定时间（令牌持有时间）过后，令牌将会传递到下一个站。 循环结束时，第一个站将再次接收令牌。

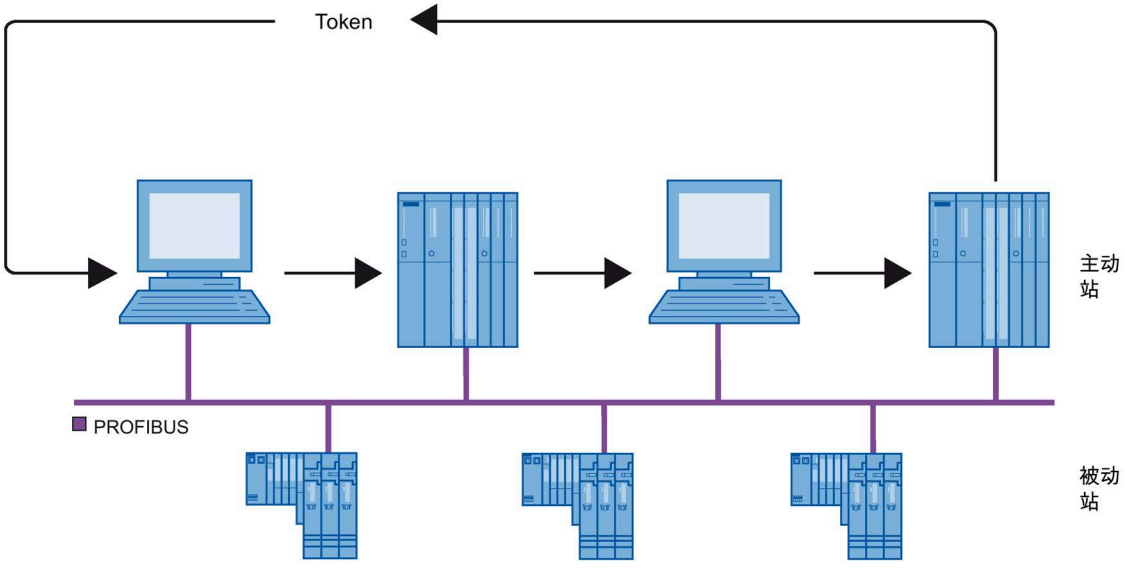

图 2-3 PROFIBUS 中的令牌传递方法

主站-从站原则是什么？

如果通信以主站/从站原则为基础，则存在称为主站的站，该站能够主动触发与从站之间的通信。 然后，从站对主站进行响应。 从站能够在其响应期间传输数据。 与主站不同，从站从不自行激活。

主站-从站原则在 PROFIBUS 中如何工作？

在 PROFIBUS 网络中，有两种基本类型的节点：

- 主动节点（主站）控制总线上的通信。每个主动节点在每个周期均接收一次令牌，然后与主动和被动节点进行通信。 令牌持有时间过后， 节点立即将令牌传递给下一个主站。 例如，DP 主站和 S7 服务器为主动站。
- 被动节点（从站）不能自行启动通信。 它们不接收令牌且仅响应来自于主动站的轮询。 被动站的典型示例为 DP 从站。

2.2.3 PROFIBUS - 其通过何种方式来适应 ISO-OSI 参考模型？

PROFIBUS 通过如下所示方式来适应 ISO-OSI 参考模型

PROFIBUS 以 ISO/OSI 参考模型为导向 但不实施所有层。 下面的示意图说明 ISO/OSI 参考模型的哪些层包含在为 PROFIBUS 所定义的各种协议中。 每个协议都为最后实施的层提供一个用户接口，通过该接口可使用数据通信服务。

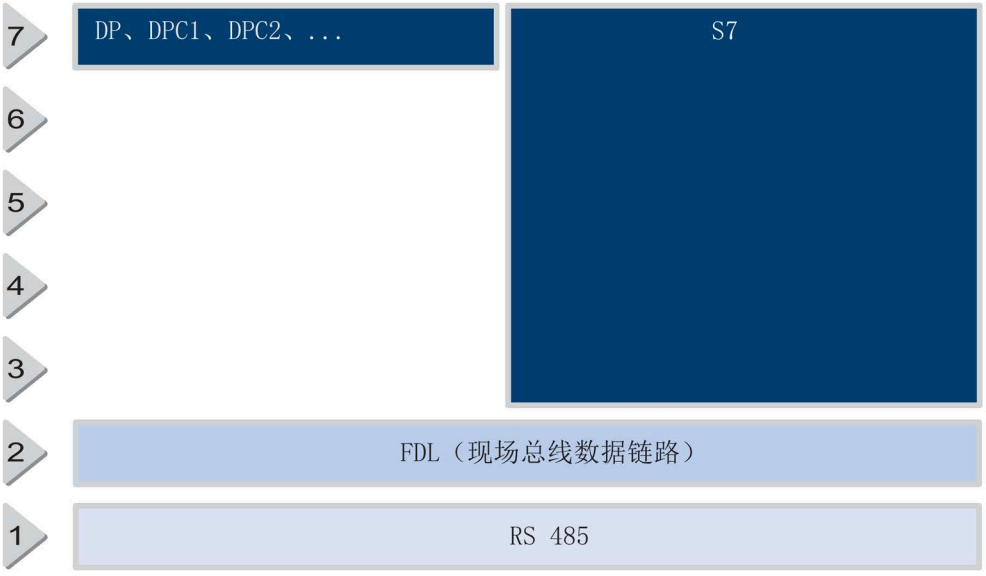

图 2-4 ISO/OSI 参考模型中的 PROFIBUS

2.3 以太网工业通信 - 概述

2.3.1 工业以太网 - 它是什么？

工业以太网定义

工业以太网是符合国际标准 IEEE 802.3（以太网）的强大通信网络，其旨在满足工业应用中的各项要求。

工业以太网的属性有哪些？

其主要特征如下：

- 为不同的的应用领域（例如管理和工厂）联网。
- 设计可靠且抗电磁干扰。
- 高传输性能（100 Mbps 和 1 Gbps）。
- 支持各种传输介质，例如双绞线或光纤电缆。
- 交换型以太网技术使性能可扩展。
- 冗余网络拓扑实现了高可用性。
- 使用各种传输协议来传输大量数据。
- 可与 PROFINET IO 进行实时传输。

工业以太网是如何构建的？

工业以太网的拓扑通常为星型结构。

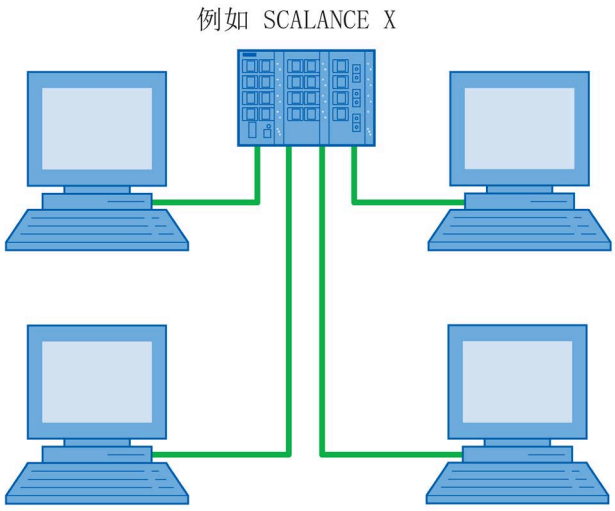

图 2-5　　工业以太网的典型拓扑

如何通过工业以太网传输大量数据？

通过以太网传输数据的一个特点是，会限制最大数据包的大小。 如果需要传输大量数据，则必须将其拆分成若干个包。 此任务由各种传输协议来处理：

ISO 传输协议支持将大量数据拆分成数据包，从而允许传输大量数据。 这与 ISO/OSI 参考模型的第 4 层相对应。

ISO on TCP 符合带有根据 ISO/OSI 参考模型的第 4 层扩展的 RFC 1006 的 TCP/IP 标准。 通过该扩展，支持将大量数据拆分成数据包，从而允许传输大量数据。 RFC 1006 为官方标准，许多制造商均对其提供支持。

原生 TCP/IP（无 RFC 1006）不支持将大量数据拆分成数据包。 因此，该任务必须由通信伙伴双方的用户程序实施。

2.3.2 交换型以太网 - 它是什么？

交换型以太网定义

交换型以太网将网络分割成由交换机链接的各个段。

交换型以太网的优势有哪些？

- 将网络分成段能够减轻网络总负载。
- 每个段均可充分利用数据传输率。
- 由于使用单独的线路进行发送和接收，因此在全双工模式下不会发生数据包之间的冲突。

2.3.3 工业以太网 - 其在 ISO/OSI 参考模型中实施哪些层？

以太网通过何种方式来适应 ISO-OSI 参考模型？

工业以太网以参考模型的各个层为基础，其提供若干用户接口，通过这些接口可使用各种协议的通信服务。

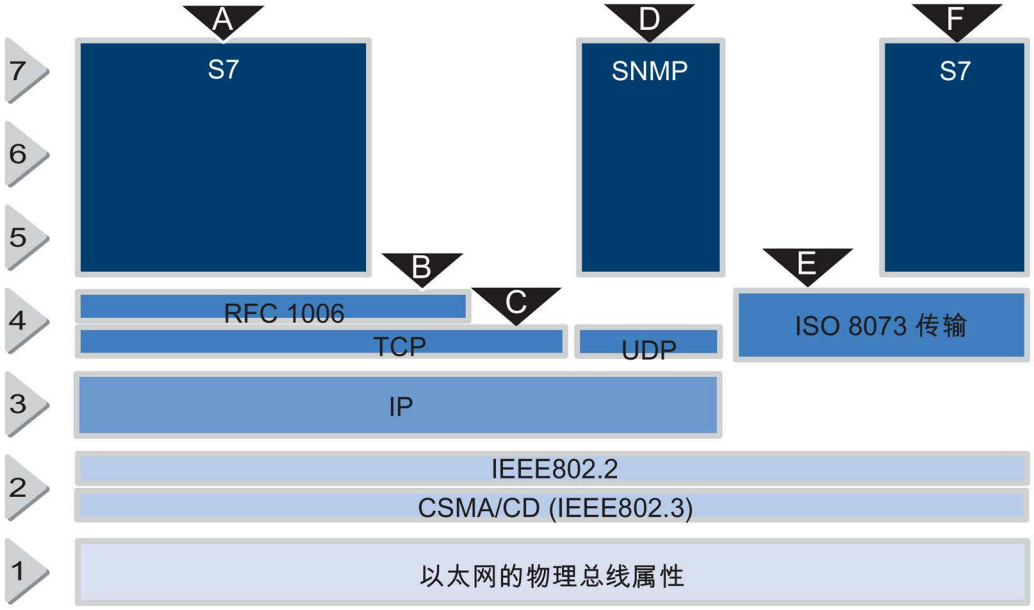

图 2-6 ISO/OSI 参考模型中的以太网

符号	协议	说明
A、F	S7 通信	使用 S7 功能的 TCP/IP (A) 和 ISO (F) 的统一用户接口
B、E	开放式通信服务 (SEND/RECEIVE)	基于 ISO 传输接口的通信服务，用来与 S5、S7 和第三方设备交换数据。 使用 TCP/IP 需要适配器 (RFC 1006)。 这便统一了 TCP/IP (B) 和 ISO (E) 的开放式 SEND/RECEIVE 用户接口。
C	原生 TCP/IP	基于 TCP/IP 的简单通信服务，用来与支持 TCP/IP 的任何设备交换数据。
D	SNMP 通信	基于 UDP/IP 的通信服务，用来与任何 SNMP 兼容设备交换数据。

2.4 PROFINET - 概述

2.4.1 PROFINET - 它是什么？

PROFINET 定义

PROFINET 代表 **PRO**cess **FI**eld **NET**，是基于工业以太网的工业自动化开放式创新标准。使用 PROFINET，可在生产自动化和运动控制中实施解决方案。 在全集成自动化 (TIA) 框架内，PROFINET 一贯地延续了已建立的 PROFIBUS 现场总线系统和工业以太网单元级的通信总线。 有了 PROFINET，可采用与基于组件的分布式自动化系统（基于组件的自动化）相同的方式，将简单分布式现场设备和时间关键应用 (PROFINET IO) 集成到以太网通信中。

PROFINET 能够满足自动化工程的所有要求。 目前，已将通过 PROFIBUS 和工业以太网获得的经验融入 PROFINET 中。 从一开始，PROFINET 的定义就由开放式标准的使用、简单处理和现有系统组件的集成所决定。

如今，PROFINET 被定义为 PROFIBUS 用户组织协会 (PNO) 的跨供应商通信、自动化和工程模型，并根据 IEC 61158 进行集成。

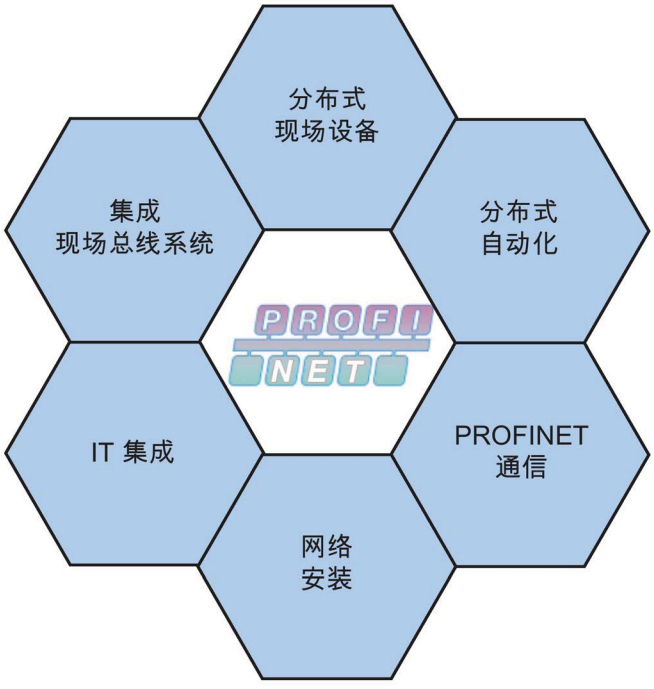

图 2-7　　符合 PNO 的 PROFINET 定义

PROFINET 的目标是什么？

PROFINET 的目标如下：

- 通过现场总线和以太网进行全集成通信
- 开放式、分布式自动化
- 实时通信
- 使用开放式标准

2.4.2 PROFINET - 它基于哪种通信？

PROFINET 基于以下通信

PROFINET 基于以太网通信。 其可扩展并提供三个性能级别：

1. TCP、UDP 和 IP，用于非时间关键数据，例如非循环读写数据记录、参数分配和组态（非实时/NRT）
2. 实时 (RT) 高性能通信，用于过程自动化中的时间关键过程数据。
3. 等时实时 (IRT) 高性能、确定式定时通信，用于运动控制中的时间关键过程数据

PROFINET 中定义了哪个实时通信？

在自动化工程中，有一些应用需要更快速的更新和响应时间。 为此，PROFINET 标准定义了实时通信机制。 如上述内容所提及，实时通信标定如下：

- **实时通道**（RT 通道）是直接基于以太网第 2 层的实时通信通道，其使用 RT 协议。 由于此解决方案忽略了某些通信级，因此通信级所需的时间降至最低。过程数据的刷新率有所提高，因为数据的准备有利于更快速地传输，并且接收数据的用户程序能够更快地处理数据。 刷新和响应时间仅 5 - 10 ms。
- **等时实时通道**（IRT 通道）专为运动控制应用而开发。 在此，要求刷新和响应时间小于 1 ms。为实现此目标，IRT 通道以快速以太网 (100 Mbps) 的第 2 层为基础，并使用 IRT 协议。 此外，数据还通过受时隙控制的传输方法来传输。 由于在以太网上通信伙伴的时间同步，因此可指定时隙，并以之将通信拆分成确定式和开放式通道。 时间关键的实时数据在确定式通道中传输，非时间关键数据则在开放式通道中传输。

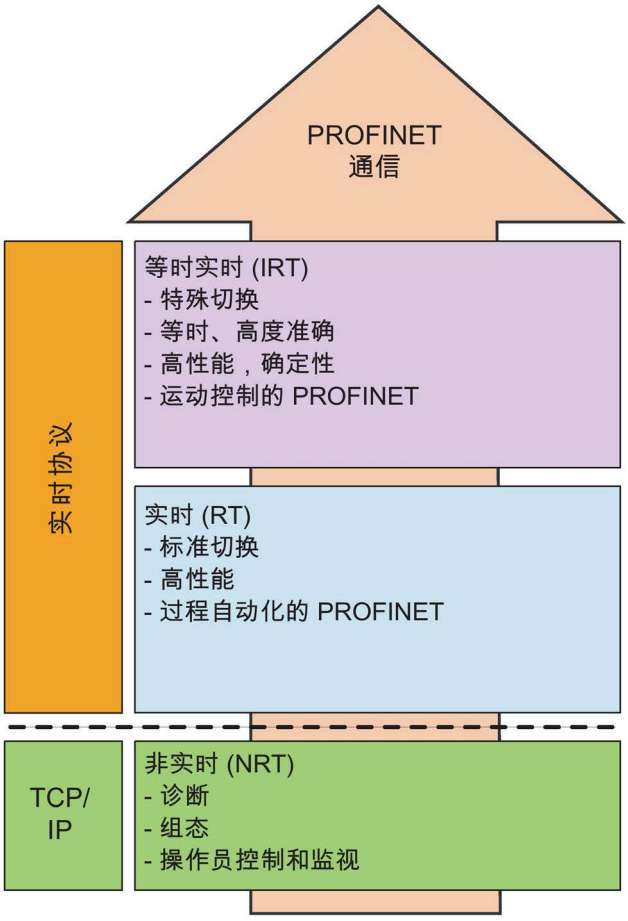

图 2-8　　SIMATIC NET 中的 PROFINET 实时通信

2.4.3 PROFINET - PROFINET 在 ISO/OSI 参考模型中实施哪些层？

ISO/OSI 参考模型中的 PROFINET

PROFINET 以参考模型的各个层为基础，有效提供了用于数据传输的两个通信通道： RT 通道和 IRT 通道。

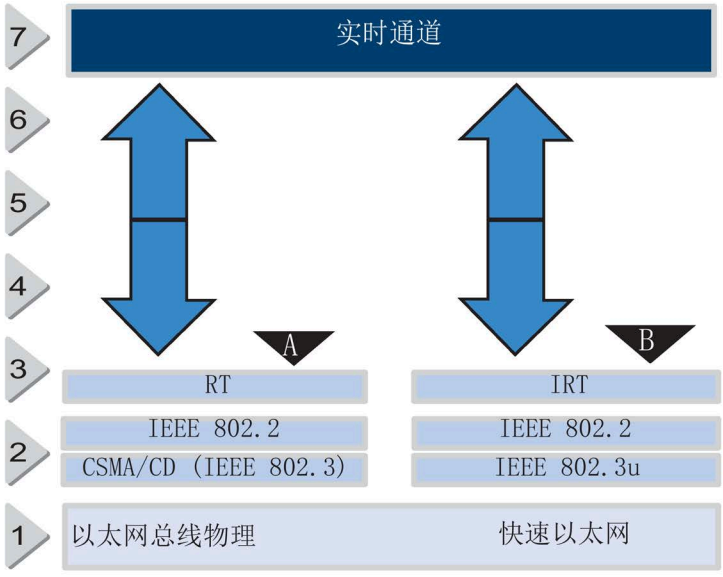

图 2-9 ISO/OSI 参考模型中的 PROFINET

符号	说明
A	RT 通道，用于与 PROFINET IO 进行通信
B	IRT 通道，用于与 PROFINET IO 进行通信

2.5 SEND/RECEIVE 协议

2.5.1 SEND/RECEIVE 协议 - 它是什么？

SEND/RECEIVE 协议

SEND/RECEIVE 协议是用于通过 PROFIBUS 和工业以太网传输数据的通信协议。它允许可编程控制器之间的简单数据交换。通过 SEND/RECEIVE 协议，SIMATIC S5 设备、SIMATIC S7 设备、PC、工作站以及第三方设备之间可以互相通信。

PROFIBUS 和以太网中的 SEND/RECEIVE 协议有何区别？

- 在 PROFIBUS 中，SEND/RECEIVE 协议基于 FDL 服务，而在以太网中，其使用传输层中的可用服务。
- 使用 PROFIBUS 可传输的数据量限制在 246 个字节内，以太网中可传输的最大数据量为 4096 个字节。
- 与以太网不同的是，PROFIBUS 没有变量服务。

2.5.2 SEND/RECEIVE 协议 - 典型系统组态外观如何？

本部分将介绍在 PROFIBUS 和工业以太网上通过 SEND/RECEIVE 协议实施不同设备间数据通信的典型系统组态。

SEND/RECEIVE 协议在 PROFIBUS 中的系统组态示例

对基于 PROFIBUS 的 SEND/RECEIVE 协议通信，SIMATIC NET 系列为 SIMATIC S5、SIMATIC 505 和 SIMATIC S7 系列控制器以及 PC、工作站和第三方设备提供了通信模块。

为此，SIMATIC S7 提供了通信模块 CP 342-5 和 CP 443-5 以及面向 PC 的模块，例如 CP 5623。

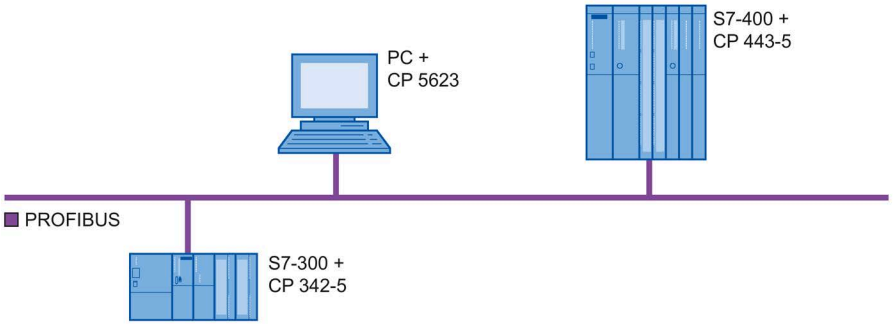

图 2-10　PROFIBUS 的典型系统组态

SEND/RECEIVE 协议在以太网中的系统组态示例

对基于以太网的 SEND/RECEIVE 协议通信，SIMATIC NET 系列为 SIMATIC S5、SIMATIC 505 和 SIMATIC S7 系列控制器以及 PC 和工作站提供了通信模块。

为此，SIMATIC S7 通常提供通信模块 CP 343-1 和 CP 443-1 以及面向 PC 和工作站的模块，例如 CP 1623。

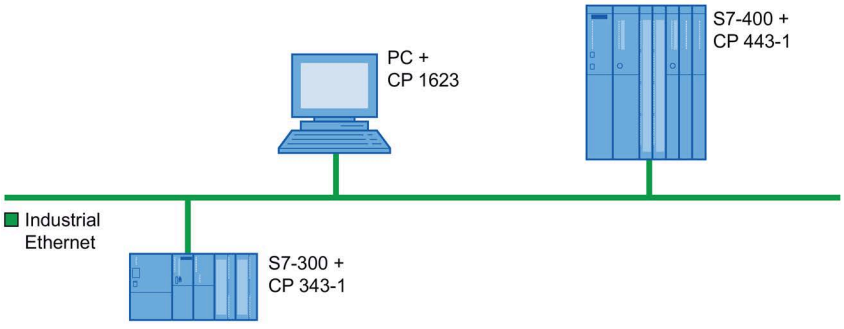

图 2-11　以太网的典型系统组态

2.5.3 SEND/RECEIVE 协议的工作原理是什么？

SEND/RECEIVE 协议与 PROFIBUS 协作的方式

用于 PROFIBUS 的 SEND/RECEIVE 协议以 FDL 数据块中数据的简单传输为基础。 这样便可直接使用 PROFIBUS 数据传输层的服务，即现场总线数据链路 (FDL)。 为了传输数据，接收方需提供接收缓冲区，发送方会将正在传输的数据写入至该缓冲区。

只能在主动 PROFIBUS 节点之间使用 SEND/RECEIVE 协议进行数据通信。 FDL 数据块的大小将用户数据限制在每帧最多 246 个字节。 数据交换利用 SDA（发送数据需要确认）和 SDN（发送数据无需确认）服务。

使用 SEND/RECEIVE 协议进行通信无需建立连接。

SEND/RECEIVE 协议与以太网协作的方式

与使用 PROFIBUS 进行数据通信不同，使用工业以太网时，SEND/RECEIVE 协议基于 ISO/OSI 参考模块的传输层。 它为用户提供传输层的服务，例如连接、流控制和数据分段。

SEND/RECEIVE 协议所用的传输协议可与工业以太网以及 ISO 传输协议和 TCP/IP 传输协议（不管有无 RFC1006）一起使用。

ISO 传输协议在国际标准 ISO 8073 第 4 类中指定，并为数据传输提供服务。
您可以选择使用数据分段，也就是说，可在 ISO 传输层上将用户数据分割成多个数据帧，之后 ISO 传输服务便能传输大量数据。 只要通信伙伴支持符合 ISO 传输的数据发送和接收，ISO 传输服务即允许与之进行通信。

ISO-on-TCP (RFC1006) 协议符合具有 RFC1006 的 TCP/IP 标准（传输控制协议/Internet 协议）。 由于 TCP 在未对包内数据进行分段的情况下实施数据通信，因此需要 RFC1006。 RFC1006 描述 ISO 服务如何传输协议以及如何藉此将数据分段映射至 TCP。 RFC1006 是许多供应商使用的官方标准。

原生 TCP/IP 协议（无 RFC1006）允许与能够使用 TCP/IP 的任何通信伙伴进行通信。 由于 TCP/IP 的传输层传送非结构化数据流，因此进行分段的任务便留给了用户。 通信连接的伙伴双方均需获知待传输数据包的大小，以便从数据流中挑选出正确的数据包。

通过以太网使用 SEND/RECEIVE 协议进行的数据通信始终面向连接。 这意味着，必须首先建立到伙伴设备的传输连接，然后才能传输数据。 建立连接时，一个伙伴为主动方，另一个则为被动方。 主动节点会启动到伙伴设备的连接建立。 两个设备中负责建立连接的设备在连接组态中进行设置。

2.5.4 SEND/RECEIVE 协议 - 提供哪些通信服务？

SEND/RECEIVE 协议提供以下通信服务

对于数据交换，SEND/RECEIVE 协议提供缓冲区发送/接收以及变量服务。 缓冲区发送/接收服务用于在两个可编程控制器之间传输非结构化数据块，且在 PROFIBUS 和以太网中均可用。 变量服务用于传输结构化数据，即在可编程控制器上定义的变量。 变量是可编程控制器中所谓的数据对象。 示例包括数据块、I/O 输入和输出、位存储器、定时器、计数器和系统区。 变量服务仅可在以太网上使用。

对于 PC，PROFIBUS 中的若干附加服务并非面向数据通信，而是用于诊断和信息收集：

- 获取总线参数和本地站地址
- 获取总线上的站列表
- 标识本地站和伙伴站

缓冲区发送/接收服务的工作原理是什么？

SEND/RECEIVE 协议的缓冲区服务包括两种通信服务：SEND 和 RECEIVE。

SEND 服务在发送数据的设备上使用。 数据的发送必须由发送方明确启动。 将要接收数据的设备必须先激活 RECEIVE 服务，然后才能准备接收。

PROFIBUS 上用于数据通信的 SEND 和 RECEIVE 通信服务是无需进行连接监视的简单服务，因而不会检测伙伴设备的故障。 此类监视只能由适当的用户程序实施，例如，通过触发数据的循环传输以及检查接收设备上的循环数据。

变量服务的工作原理是什么？

SEND/RECEIVE 协议的变量服务包括两种通信服务：FETCH 和 WRITE。 这些通信服务仅在以太网上可用。

执行 FETCH 服务时，作业将从 PC 发送到请求特定变量当前值的伙伴设备。 该伙伴设备以包含所要求变量当前值的数据块确认该作业。

使用写入服务，PC 能够将特定变量的当前值发送到伙伴设备。 伙伴设备对该信息进行评估并将变量设置为传输的值。 随后伙伴设备确认该服务。

2.5.5 SEND/RECEIVE 协议 - 如何组态？

SEND/RECEIVE 协议的组态方式如下

为了通过 SEND/RECEIVE 协议进行通信，必须先组态连接，然后才能使用它们。 为此，提供了“SIMATIC STEP 7 Professional”组态工具。 已组态的连接将通过在组态过程中指定的唯一连接名称进行标识。 对于 SEND/RECEIVE 协议，有四种预定义的连接类型还描述了下列连接类型：

- FDL 连接： 通过 PROFIBUS 进行的连接
- ISO 传输连接： 采用 ISO 传输协议并通过以太网进行的连接
- ISO-on-TCP 连接： 采用 ISO-on-TCP 协议并通过以太网进行的连接
- TCP 连接： 采用原生 TCP/IP 协议并通过以太网进行的连接

每个已组态的连接都必须进行参数设置。 创建连接后，组态工具将为这些参数设置默认值，用户可不作任何修改即采用这些值。 例如，这些参数包括：

- 通信伙伴的地址
- 服务访问点 (SAP)。

2.5.6 SEND/RECEIVE 协议 - 有哪些优缺点？

PROFIBUS 中 SEND/RECEIVE 协议的优点如下

PROFIBUS 中的开放式 SEND/RECEIVE 协议具有以下优点：

- 最多可传输 246 个字节的大型数据块。
- 不传输数据时，将不会有网络负载。
- 可将广播帧发送至多个节点。
- 可在 PC 上对数据块进行结构化访问。
- 可与 SIMATIC S5 和 SIMATIC S7 设备进行通信。
- PC/PG 可彼此进行通信。

PROFIBUS 中 SEND/RECEIVE 协议的缺点如下

PROFIBUS 中的 SEND/RECEIVE 协议具有以下缺点：

- 接收器不能启动数据传输。 它必须一直等到发送器传输数据。
- 不会进行任何监视来检测接收器故障或网络中断。
- 没有路由（将作业转发到其它网络）。

以太网中 SEND/RECEIVE 协议的优点如下

以太网中的 SEND/RECEIVE 协议具有以下优点：

- 可通过分片来传输多达 64 KB 的更大的块数据。
- 如果用户未启动任何数据传输，则不会有网络负载。
- 可对数据块进行结构化访问。
- 可与 S5 和 S7 设备以及 PC 进行通信。
- 通过变量服务可灵活访问数据。

以太网中 SEND/RECEIVE 协议的缺点如下

以太网中的开放式 SEND/RECEIVE 协议具有以下缺点：

- 接收器不能启动数据传输。 它必须一直等到发送器传输数据。
- 数据必须由伙伴设备上的用户程序放置或复制到缓冲区。
- 使用变量服务时的数据吞吐量低于使用缓冲区发送/接收服务时的数据吞吐量。
- 要监视变量更改，必须周期性访问伙伴设备，并涉及更高的网络负载。

2.6 DP 协议

2.6.1 DP 协议 - 它是什么？

DP 协议

DP 协议用于分布式外围设备 I/O (DP)，可实现在过程临近区域使用一些模块和其它现场设备。 它基于现场区域的通信标准 (IEC 61158)，并在 PROFIBUS 标准 (EN 50170) 中进行指定。

在 PROFIBUS 上使用 DP 协议，可覆盖 I/O 设备间的长距离。 分布式 I/O 站在本地收集输入信号并将其设置为可用，以便信号能够被获取。 然后，计算机上的 CPU 便可周期性地获取它们。 中央控制器以相反方向将输出数据周期性地发送至分布式 I/O 站。

DP 协议专用于时间关键型应用。 简单的优化传输协议、高传输率以及主站-从站原则的使用可实现较短的循环时间。

DP 协议的属性有哪些？

属性有：

- 由主站进行中央控制
- 使用简单传输协议实现大数据吞吐量
- 以输入和输出方向循环传输过程映像
- 以在线诊断检测错误
- 由于它基于 ISO/OSI 参考模型的 PROFIBUS 第 2 层，因此能够与其它设备（主站和从站）并行操作。

为 DP 协议定义了哪些扩展？

以下部分提供各种 DP 主站及其扩展的概述。

对于 DP 主站，为周期性数据交换和诊断功能定义 1 类和 2 类。 此外，还为非周期性通信实施扩展 C1 和 C2。

什么是 DPV1？

DPV1 标准代表 DP 通信的扩展。 支持 DPV1 的从站具有额外的存储区域，可在此区域中存储从站特定的特殊数据记录。 DPV1 包括两个部分，一个是循环主站的 DPC1 扩展，另一个是附加诊断和参数分配功能的 DPC2 扩展。 假设功能得到扩展，使用 DPV1 功能能够读取或写入数据记录。

什么是 1 类 DP 主站？

1 类 DP 主站提供用于将参数分配到从站以及周期性数据交换的服务。

什么是 DPC1？

DPC1 是 1 类 DP 主站的 DPV1 扩展。 它使 C1 主站能够非周期性地读取和写入 DPV1 从站的附加数据区域。

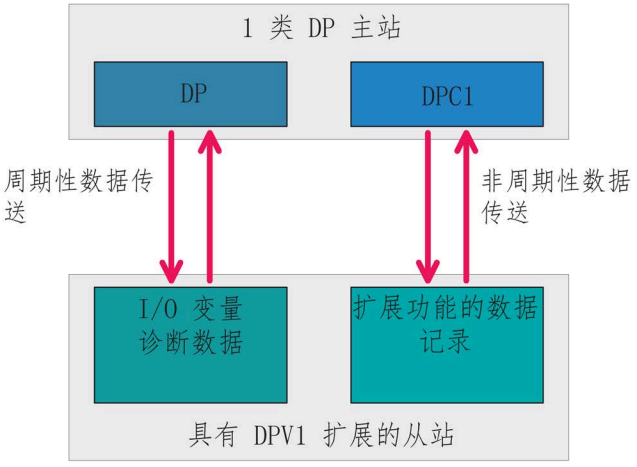

图 2-12　　1 类 DP 主站和 DPC1

什么是 2 类 DP 主站？

2 类 DP 主站提供诊断选项，并可在不干扰网络运行的情况下查询 1 类 DP 主站或 DP 从站的状态。

什么是 DPC2？

DPC2 是 2 类 DP 主站的 DPV1 扩展。 它使 C2 主站能够非周期性地读取和写入 DPV1 从站的附加数据区域。

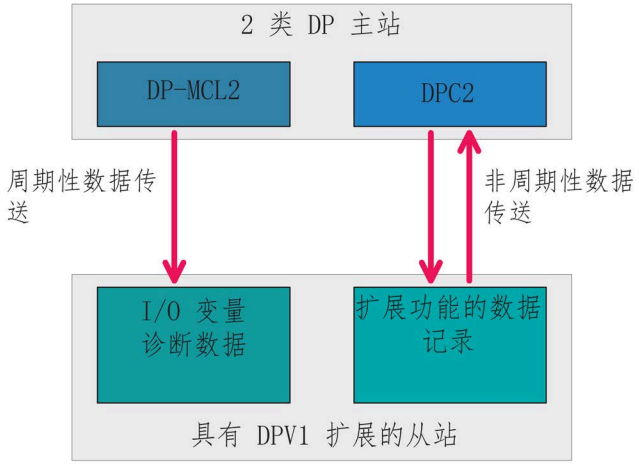

图 2-13　　2 类 DP 主站和 DPC2：

2.6.2 DP 协议 - 典型系统组态的外观如何？

以下部分介绍了在不同设备之间使用 DP 协议实施数据通信的 PROFIBUS 中的典型系统组态外观。

DP 协议的系统组态示例

对基于 PROFIBUS 的 DP 协议通信，SIMATIC NET 系列为 SIMATIC S5 和 SIMATIC S7 系列控制器和 PC 与工作站都提供了通信模块。

SIMATIC S7 可用的典型通信模块为 CP 343-5，PC 和工作站可用的典型通信模块为 CP 5623 或 CP 5622 等。ET 200 系列中还具有与 DP 兼容的模块。

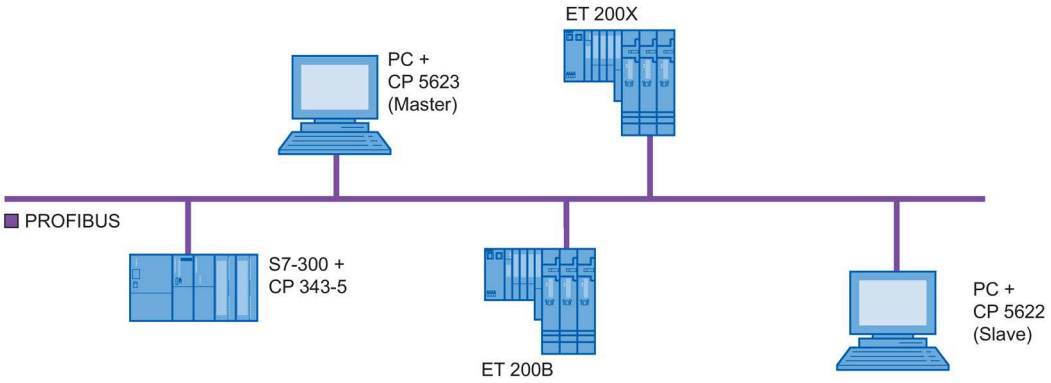

图 2-14　PROFIBUS 的典型系统组态

2.6.3 DP 协议 - 工作原理是什么？

DP 协议工作原理

分布式 I/O 系统中有三种类型的通信伙伴：

DP 通信伙伴	说明
DP 从站	被动总线节点，通常为 I/O 设备。 主动总线节点，例如用于处理附加任务的 PG 通信 CP。
1 类 DP 主站	主动节点，控制 DP 从站的中央组件。
2 类 DP 主站	可用于与 1 类主站并行调试和诊断的主动节点。

DP 主站和分布式 I/O 站之间的正常通信采用轮询的形式。 轮询是指 DP 主站将周期性轮询帧发送给已为其分配的每一个 DP 从站。

轮询帧包含将由 DP 从站应用到其输出端口的当前输出数据。 DP 从站通过返回确认帧来确认接收。确认帧包括在 DP 从站输入端口应用的输入数据。
如果 DP 从站没有输出或输入端口，则将发送“空帧”作为替代。

在一个轮询周期中处理所有运行中的 DP 从站。 在处理了最后一个从站后，将立即启动下一个轮询周期。 此方法能够确保日期是最新的。 在每个轮询周期中，DP 主站都尝试在周期中包含非运行中的从站。

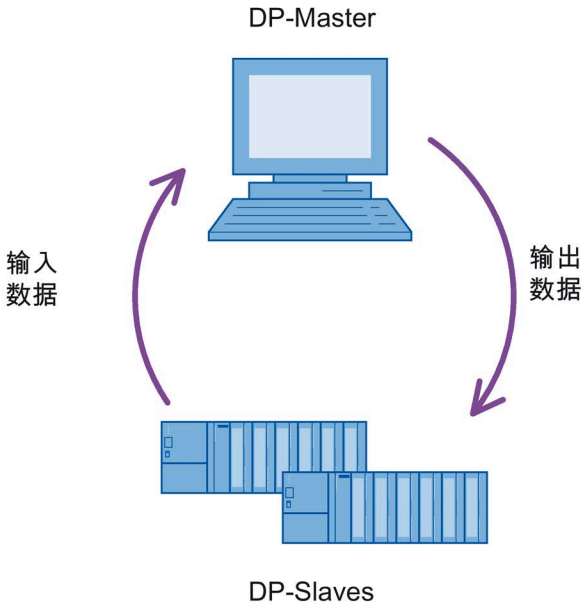

图 2-15　　DP 主站与 DP 从站之间的通信

已为主站与从站之间的快速数据吞吐量优化了 DP 协议，因而该协议不包括流控制。

2.6.4　　DP 协议 - 如何组态？

DP 协议的组态方式如下

对于与 DP 协议的通信，必须先为 DP 设备分配参数并将其组态，然后才能开始生产运行。 为此，提供了“SIMATIC STEP 7 Professional”组态工具。 通过选择协议，也可为每个 DP 设备选择全局操作参数作为参数分配数据，并选择输入/输出端口的数量和类型作为组态数据。

2.6.5　　DP 协议 - 有哪些优点？

DP 协议的优点如下

DP 协议具有下列优点：

- 通过 PROFIBUS 进行的通信的效率非常高且具有实时功能。
- 用户程序可对过程数据进行简单而快速的访问。
- DP 协议是得到广泛应用且国际标准化的开放式协议。

2.6.6 1 类 DP 主站 - 哪些通信服务可用？

1 类 DP 主站具有以下可用模式

DP 主站控制 DP 系统的状态。 DP 主站的每个模式的特征都是 DP 主站与 DP 从站之间的操作已定义：

模式	含义
OFFLINE	DP 主站和 DP 从站之间无论如何都不会存在任何 DP 通信。 这是 DP 主站的初始状态。
STOP	在此模式下，DP 主站与 DP 从站之间同样不存在任何 DP 通信。 与 OFFLINE 模式不同，DP 诊断站（2 类 DP 主站）可从 DP 主站读取诊断信息。
CLEAR	DP 主站为 DP 从站提供其启动时所需使用的数据（参数分配和组态）。 随后，在 CLEAR 模式下，值 0h 将被发送到具有过程输出的所有从站；换言之，过程输出处于安全状态。 从站的输入数据已知并可读取。
OPERATE	DP 主站和 DP 从站之间存在周期性数据传输。 这称为生产阶段。 在此模式下，由 DP 主站对 DP 从站进行相继轮询。

从当前模式开始，必须以选定顺序（升序或降序）OFFLINE - STOP - CLEAR - OPERATE 运行模式。

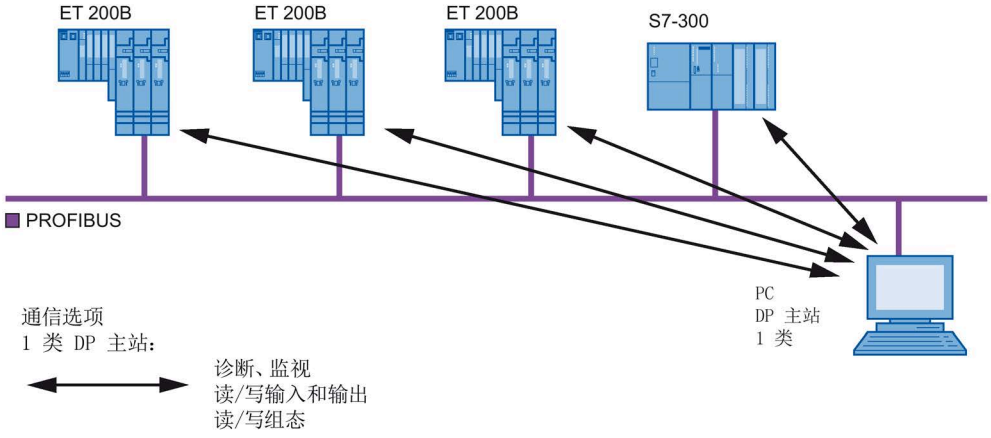

图 2-16 1 类 DP 主站的通信服务

1 类 DP 主站提供以下通信服务

不直接应用 1 类 DP 主站来访问过程变量，而是经由通信模块上的过程映像进行访问。

过程映像包括每个 DP 从站的三个数据区：

- 来自 DP 从站的输入数据
- 发送至 DP 从站的输出数据
- 来自 DP 从站的诊断数据

PC 上的用户程序可通过 1 类 DP 主站使用以下服务：

- DP 主站过程映像的变量服务

以下信息服务同样可用：

- DP 主站和 DP 从站的模式
- 来自 DP 主站的事件消息
- 由 P 模块进行的活动监视
- DP 从站的类型

1 类 DP 主站有哪些优缺点？

使用 1 类 DP 主站具有以下优点：

- 快速访问周期性数据。
- 由于可直接从过程映像获取数据，且不会显式导致通信，因此必须极快地处理来自应用程序的作业。

使用 1 类 DP 主站具有以下缺点：

- 周期性交换输入和输出数据会导致总线负载非常高。

2.6.7 2 类 DP 主站 - 哪些通信服务可用？

2 类 DP 主站的工作原理

除属于 1 类 DP 主站的设备外，DP 系统还可包含 2 类 DP 主站设备。 这些设备用于调试、组态或诊断。

例如，可将 2 类 DP 主站连接到 PROFIBUS 以用于诊断目的。 然后，便可在不干扰网络运行的情况下随时查询从站和 1 类主站的状态。 如果从站允许这样做，则 2 类 DP 主站也可更改从站地址。

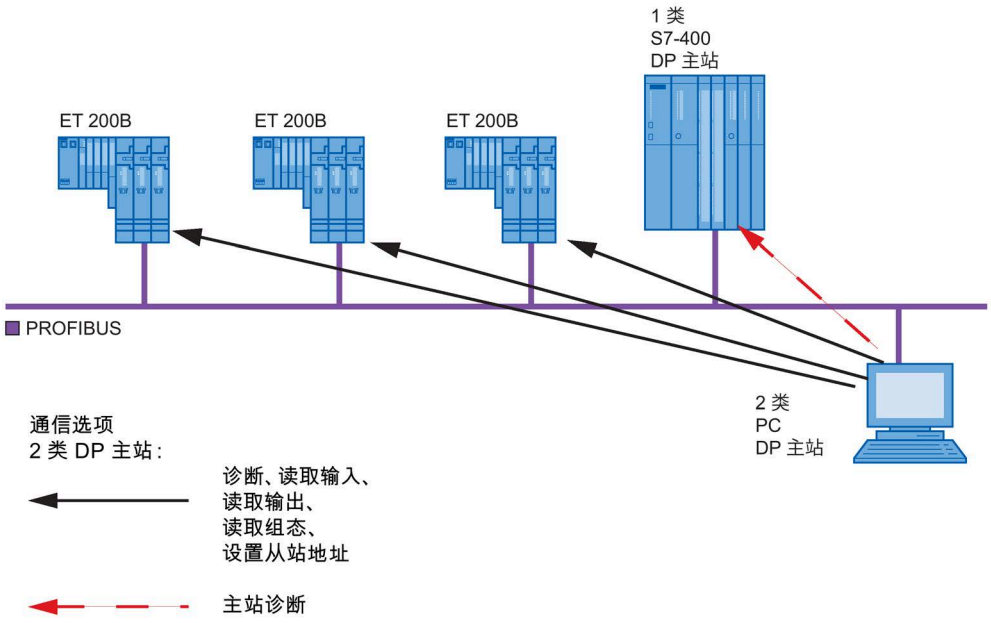

图 2-17 2 类 DP 主站的通信服务

2 类 DP 主站提供以下通信服务

与 1 类 DP 主站的运作方式相同，2 类 DP 主站也能够访问周期性输入和输出数据并将信息映射到变量。 但是，该数据只能读取却不能写入。

2 类主站的基本功能如下：

- 读取从站的数据
- 读取 1 类主站的数据

2 类 DP 主站有哪些优缺点？

使用 2 类 DP 主站具有以下优点：

- 对网络内运行的影响程度非常有限。
- 无法修改从站地址。

使用 2 类 DP 主站具有以下缺点：

- 只能读取从站的输入、输出和诊断数据。
- 无法使用 OPC 来通过 DP 循环同步访问过程变量。

2.6.8 DPC1 - 哪些通信服务可用？

使用 DPC1 服务进行通信的工作原理

使用 DPC1 服务，除能够通过 DP 主站接口进行周期性轮询外，还能非周期性轮询从站中的数据。 每个具有 DPV1 扩展的 DP 从站都具有附加数据区，而该数据区可由 DPC1 主站进行读写。 此数据区取决于特定从站，并可包含参数分配数据或报警消息。 通过指定插槽和索引处理附加数据区的各数据记录。

由于已将主站的轮询周期作为隐式连接进行启动，因此与从站的通信连接不必使用 DPC1 服务。 当具有 DPC1 功能的从站已组态并已设置了参数后，即可由 DPC1 服务进行处理。

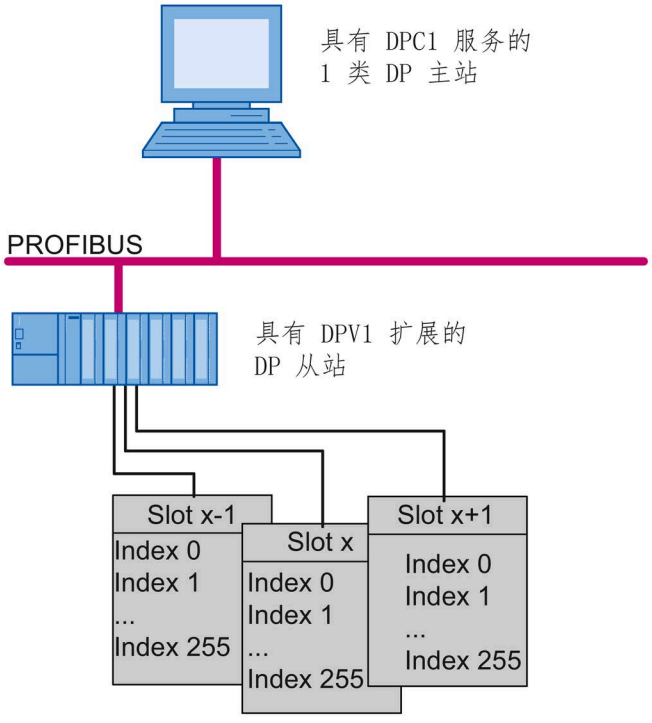

图 2-18 使用 DPC1 服务进行通信的原则

存在哪些 DPC1 通信服务？

DPC1 服务包含以下内容：

- 非周期性读写数据记录。
- 报警服务。

DPC1 服务有哪些优点？

使用 DPC1 服务具有以下优点：

- 通过非周期性访问减少总线负载。
- 可传输多达 240 个字节的数据块。
- 可对数据字段进行结构化访问。
- 可与 1 类 DP 主站的变量服务并行使用。

2.6.9 DPC2 - 哪些通信服务可用？

使用 DPC2 服务进行通信的工作原理

使用 DPC2 服务，除能够进行周期性轮询外，还能轮询 2 类 DP 主站中从站的非周期性数据。 与 DPC2 兼容的从站具有附加数据区，而该数据区可使用 DPC2 服务进行读写。 此数据区取决于特定从站，并可包含参数分配数据或报警消息。 通过指定插槽和索引处理附加数据区的各数据记录。

这种通信与正常主站-从站通信的本质区别在于，必须首先建立连接，然后再维持连接状态，直到外部影响将其中断或者主站将其终止。 只要建立此连接，主站即可与从站进行通信。

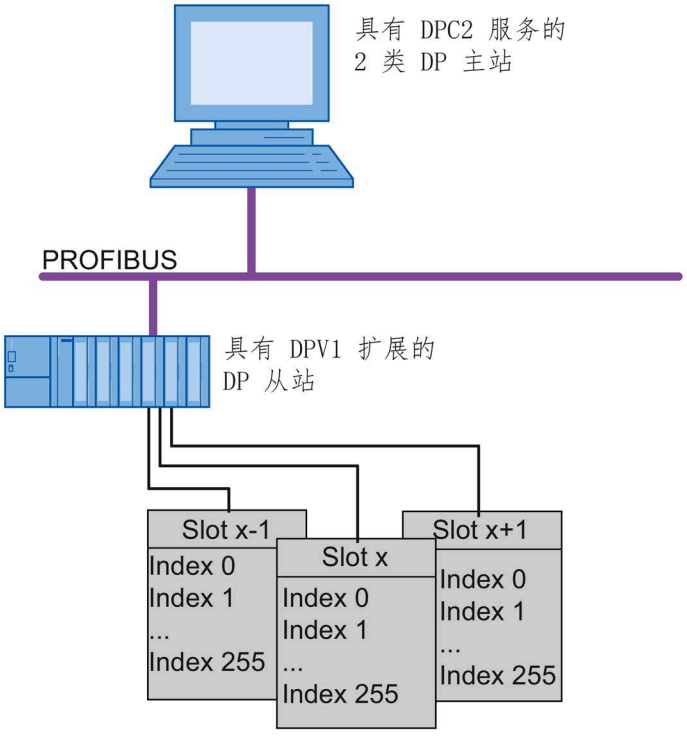

图 2-19 使用 DPC2 服务进行通信的原则

存在哪些 DPC2 通信服务？

最重要的 DPC2 服务如下：

- 通信关系的建立和终止
- 读取从站数据记录

使用 DPC2 服务可为用户提供缓冲区发送/接收服务。

由于系统以低于周期性数据服务的优先级处理 DPC2 服务，因此数据吞吐量较低。 通常，往返时间也会因网络具有附加负载而增加。

通过与 1 类 DP 主站 (DPC1) 相同的方式，2 类主站仅可将 DPV1 插槽的数据作为整个块进行访问。 读取作业将返回由插槽和索引所标识的数据记录的整个内容，写入作业将覆盖整个数据记录。

DPC2 服务有哪些优点？

使用 DPC2 服务具有以下优点：

- 可异步访问从站。
- 可传输更大的数据块。
- 可对数据字段进行结构化访问。
- 可与 1 类主站并行使用。

2.6.10 DP 从站 - 哪些通信服务可用？

DP 从站提供以下通信服务

DP 从站提供数据通信服务，从而允许 DP 主站在轮询周期期间通过 PROFIBUS 获取输入数据，以及接收和处理由 DP 主站发送的输出数据。 DP 从站还能设置可由 DP 主站读取的诊断数据。

在 DP 通信内，将 DP 从站视为模块化从站。 每个从站均可由多个子模块构成，而每个子模块又都具有其自己的输入和输出区。

适合 DPV1 扩展的从站可包含每个模块的附加数据记录。 这些数据记录包含可由 DPC1 主站读写的从站特定数据。 每个数据记录可拥有多达 240 个字节的有效负载数据。

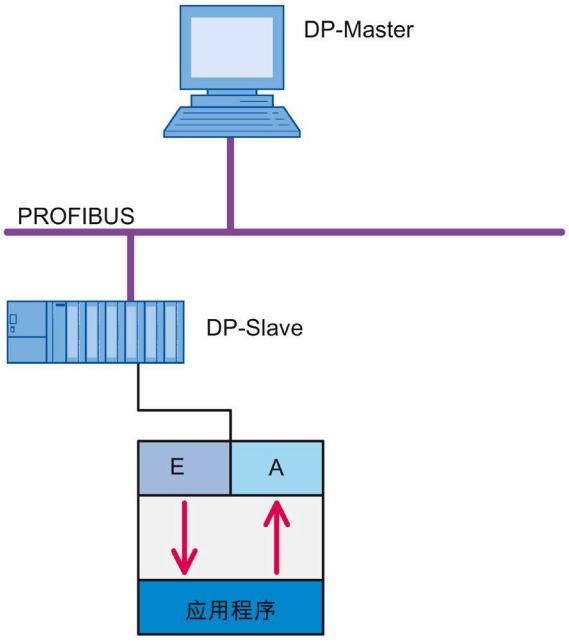

图 2-20 使用 DP 从站进行通信的原则

DPV1 DP 从站扩展具有哪些优点？

DPV1 扩展具有下列优点：

- 可进行非周期性访问。
- 可传输大型数据块。
- DP 从站具有附加诊断数据记录。

2.7 S7 协议

2.7.1 S7 协议 - 它是什么？

S7 协议

S7 协议用于与 SIMATIC S7 可编程控制器 (PLC) 进行通信。它既支持 PG/PC 与可编程控制器之间的通信，又支持 SIMATIC S7 系统中可编程控制器之间的数据交换。

S7 协议有哪些属性？

S7 协议的主要特征如下：

- 已针对 SIMATIC 通信进行了优化
- 与数据通信方面的其它自动化协议相比速度较快。
- 在管理方面可支持总线系统，单元级可支持工业以太网，现场级可支持 PROFIBUS。
- 还可与容错连接配合使用。

面向 PROFIBUS 和面向以太网的 S7 协议有何区别？

在 PROFIBUS 中，S7 协议基于 FDL 服务；而在以太网中，其使用传输层中的可用服务。S7 协议隐藏了此差异，所以用户在通信时不会察觉到任何区别。

S7 协议在 PROFIBUS 中和以太网中有哪些共同特性？

对于两个通信系统，S7 协议都提供面向连接的协议的各项优势，例如连接监视。 S7 协议中实施的所有通信服务也可无限制地提供。

2.7.2 S7 协议 - 典型系统组态的外观如何？

本部分将介绍在 PROFIBUS 上和在工业以太网上通过 S7 协议实施不同设备间数据通信的典型系统组态。

S7 协议在 PROFIBUS 中的系统组态示例

对基于 PROFIBUS 的 S7 协议通信，SIMATIC NET 系列为 SIMATIC S7 系列控制器和 PC 与工作站都提供了通信模块。

为此，SIMATIC S7 通常提供通信模块 CP 342-5、CP 343-5 和 CP 443-5 以及面向 PC 和工作站的模块，例如 CP 5623、CP 5624 或 CP 5622。各种支持 S7 协议的模块类型也可用于 ET 200 系统。

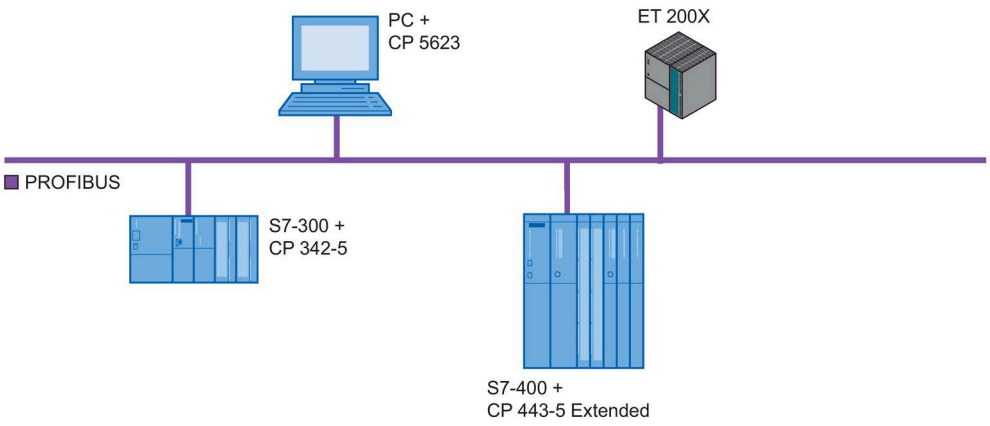

图 2-21 PROFIBUS 的典型系统组态

S7 协议在以太网中的系统组态示例

对基于以太网的 S7 协议通信，SIMATIC NET 系列为 SIMATIC S7 系列控制器和 PC 与工作站都提供了通信模块。

面向 SIMATIC S7 的典型通信模块为 CP 343-1 和 CP 443-1，面向 PC 和工作站的为 CP 1623。

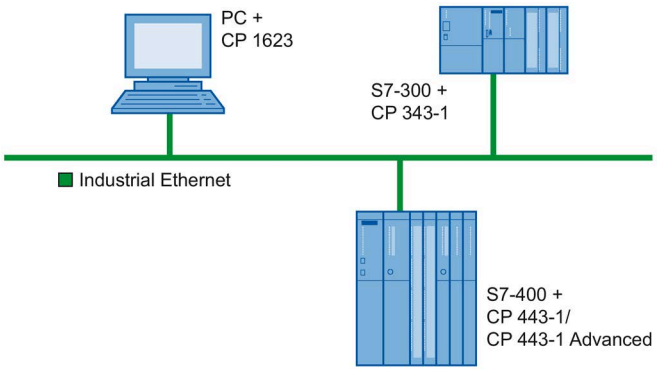

图 2-22　　以太网的典型系统组态

2.7.3　　S7 协议 - 工作原理是什么？

S7 协议工作原理

S7 协议提供简单却强大的通信服务。 根据客户端-服务器模型，在 SIMATIC PC 站的自动化应用程序与其它自动化设备之间进行数据传送。 客户端请求的数据由服务器提供。

除此之外，两个自动化设备之间也可以交换数据。 此通信也根据客户端-服务器模型来工作。

建立连接期间，通信伙伴自动调整通信路径的重要特性，以实现互相匹配。 各种设置经过协商，确保两个通信伙伴都能实现相关设置。

协商期间将确定以下事项：

- 要传送的数据包大小
- 可同时使用的发送和接收资源的数量

2.7.4 S7 协议 - 提供哪些通信服务？

S7 协议提供以下通信服务

S7 协议支持以下通信服务，这些服务可在 PROFIBUS 和以太网上无限制地提供。

通信服务	说明
信息服务	有关连接状态的信息。 显示通信伙伴的设备和用户状态。
变量服务	读取和写入一个或多个变量的功能。
缓冲区发送/接收服务	程序控制的大块数据传送。
块管理服务	凭借这些服务可在运行过程中按照可编程控制器的程序顺序对块进行下载、上传、删除和链接。允许动态修改程序顺序和参数。
事件服务	这些服务用于接收和处理来自 SIMATIC S7 可编程控制器的消息，例如报警。
安全服务	通过设置密码对 SIMATIC S7 数据对象提供访问控制。
服务器服务	PC 站点成为 S7 服务器，并提供可在本地和远程读取和写入的数据块 (PUT / GET)。 数据块的编号可在本地和远程进行查询和显示。

S7 信息服务有何用途？

通过 S7 协议提供的信息服务，

- 可以查询伙伴设备的属性。
- 可以读取伙伴设备的状态。

S7 变量服务有何用途？

通过 S7 协议可轻松访问 S7 变量。

多数 S7 设备都提供以下 S7 变量：

- 数据块
- 背景数据块
- 输入/输出
- 外设输入/输出
- 存储位
- 定时器
- 计数器

S7 变量服务有哪些优缺点？

使用变量服务具有以下优点：

- 访问伙伴设备非常轻松，无需对伙伴进行编程。
- 已优化对多个变量和较长变量数组的读取和写入。
- 在使用 OPC 服务器时，可通过分配访问权限来保护与安全相关的变量。
- 对于 OPC，可以使用 STEP 7 中的符号。
- 当使用 OPC 时，变量的大小和数据块的大小被限为最多 64 KB。

使用变量服务具有以下缺点：

- 要监视变量更改，必须周期性访问伙伴设备。
- 以较短间隔进行访问意味着网络负载较高。

S7 缓冲区发送/接收服务有何用途？

S7 缓冲区发送/接收服务允许以程序控制的方式传递较大的数据块。 最多可传送 65534 个字节的数据。

必须先组态连接，然后才可进行数据交换。 这一点不仅适用于 PC 与可编程控制器之间的连接，同时也适用于可编程控制器与可编程控制器之间的连接。

S7 缓冲区发送/接收服务有哪些优缺点？

使用 S7 缓冲区发送/接收服务具有以下优点：

- 较大（最多 65534 字节）的数据块也可传送。
- SIMATIC PC 既可以是客户端也可以是服务器，即，对于缓冲区发送/接收服务，数据也可通过 S7 协议从 PC 传送到 PC。
- 可以按 OPC 项来编排数据块的结构。
- 对于接收缓冲区内定义的所有 OPC 变量，在数据块抵达并且相应的数据已更改时都将收到一个更改消息。
- 如未发送任何数据，便不会有任何轮询方面的网络负载。

使用面向缓冲区的服务具有以下缺点：

- 发送和接收块必须在可编程控制器上以及（在适用时）在 PC 上编程。
- 接收器无法请求数据，必须等待数据发送。
- 所有 S7 可编程控制器均不提供缓冲区发送/接收服务。

S7 块管理服务有何用途？

使用 S7 协议的 S7 块管理服务提供以下应用：

- 将数据从 PG/PC 下载至 SIMATIC CPU。
- 将数据从 SIMATIC CPU 上传至 PG/PC。
- 按 SIMATIC CPU 的程序顺序链接各个块。
- 删除块。
- 压缩可编程控制器的存储器。

块代表可编程控制器上的可加载区域。 块管理服务可与组织块 (OB)、函数块 (FB)、函数 (FC)、数据块 (DB) 和系统数据块 (SDB) 配合使用。

例如，块可通过 S7-OPC 应用程序从 S7 CPU 上传到 PC，反之亦然。在 PC 上，块存储在文件中。
块名称在 S7-CPU 内唯一。 数据的最大大小限制取决于具体的 CPU。 因此，块被分割为连续传送的各个单独的段。

传送到可编程控制器的块将存储在缓冲区中。 这表示，该块尚不可供 S7 程序使用。 尽管该框可在数据块列表中可见，可通过 STEP 7 的在线功能查看，但是无法打开该块。 只有在该块已链接到激活块列表后，才可打开。

S7 块管理服务的应用示例

通过 STEP 7 编程或创建的块在调试过程中从编程设备传送到可编程控制器。 这些块（在本例中，DB_red_car.dbf、DB_green_car.dbf ...）将在程序存储器中存储为 DB1、DB2 ...。 S7-OPC 应用程序可在运行过程中上传这些块，并可在本地将其另存为文件 DB_red_car.dbf、DB_green_car.dbf...。 因而 PC 控制器可以下载和删除这些块，并影响程序的执行，例如可通过将 DB1 (DB_red_car.dbf) 替换为另一个数据块 (DB_blue_car.dbf) 来使该块成为 DB1。

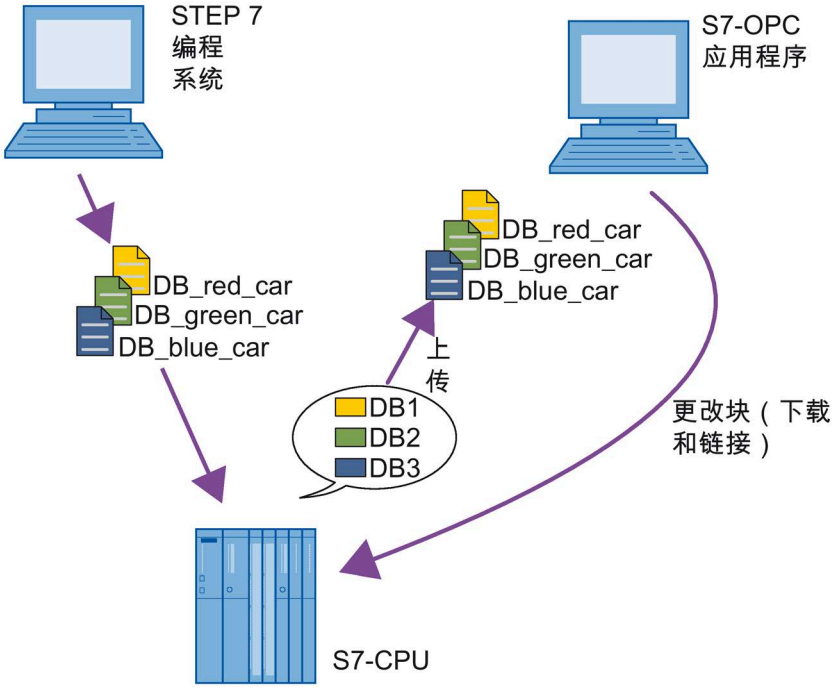

图 2-23　　S7 块管理服务示例

S7 块管理服务有哪些优缺点？

使用 S7 块管理服务具有以下优点：

- 访问（读取/写入/删除）自动化系统的可加载区域。

使用 S7 块管理服务具有以下缺点：

- 数据的大小受具体 CPU 的限制，因此数据需要连续传送。

S7 报警服务有何用途？

通过 S7 报警服务可接收可编程控制器的报警。 例如，使用该服务可发送干扰和故障等信号。

要在 PC 上详细显示报警，它们可最多连同 10 个关联值一同发送。 报警为异常事件，由可编程控制器触发。 它们将进行缓冲并且不会丢失。

S7 报警服务有哪些优缺点？

使用 S7 报警服务具有以下优点：

- 消息将进行缓冲并且不会丢失。
- 一个报警可最多连同 10 个关联值进行传送。

使用 S7 报警服务具有以下缺点：

- 必须在控制器中创建程序，才能生成消息。
- 相关值仅支持有限数量的数据类型。

S7 安全服务有何用途？

S7 安全服务管理对 S7 连接的访问。 利用它可以传送合法授权密码，从而在一个连接上取消一个保护级别。

使用 STEP 7 组态工具可为 S7 自动化系统的块管理服务激活三个保护级别：

- 基于钥匙开关设置的保护
- 写保护
- 写入和读取保护

通过传送正确的密码，可取消当前连接的所有上述这些保护级别。

S7 安全服务有哪些优点？

使用 S7 安全服务具有以下优点：

- 对连接的访问控制
- 可取消基于钥匙开关的访问控制

PC 站的 S7 服务器服务有何用途？

PC 站将成为 S7 服务器：

- 提供一个大小为 65535 个字节的数据块 DB1。
- 伙伴（例如 S7 站或 PC 站）可通过 PUT 和 GET S7 服务读取或写入数据块的值。
- PC 站的客户端可通过 S7 连接"@LOCALSERVER"读取或写入数据块的值。
- 可在存在同时访问时保证数据的一致性。
- 即使在 PC 站重启后，数据块中的值也将保留（永久性数据）。
- 数据块的编号可在本地和远程进行查询和显示。

S7 服务器服务的应用示例

例如，S7 客户端可以是一个要向 PC 站报告状态数据而且不需要站点持续对状态数据进行轮询的 S7-200 站。 然后其会将状态数据写入到数据块（不定期）。 可通知 PC 站的本地客户端状态值更改的信息。 因而，可避免对 S7 站上状态值的持续轮询。

S7 服务器服务有哪些优缺点？

使用 S7 服务器服务具有以下优点：

- 使 S7 站不必再在监视变量更改时对数据进行周期性访问。
- 数据一致性和性能。

使用 S7 服务器服务具有以下缺点：

- 只有一个数据块可用，无存储位、输入、输出、计数器或定时器。
- 在必要时，传递数据必须使用 S7 应用程序。

要激活 S7 服务器服务，PC 站上必须有一个本地客户端已激活。

2.7.5 什么是容错 S7 连接？

容错 S7 连接是一些经过组态的特殊 S7 连接，可通过工业以太网将 PC 站连接到容错 S7-400H 自动化系统，也可用于连接此类自动化系统（见下图）。 利用容错 S7 连接无法将 PC 站连接在一起。

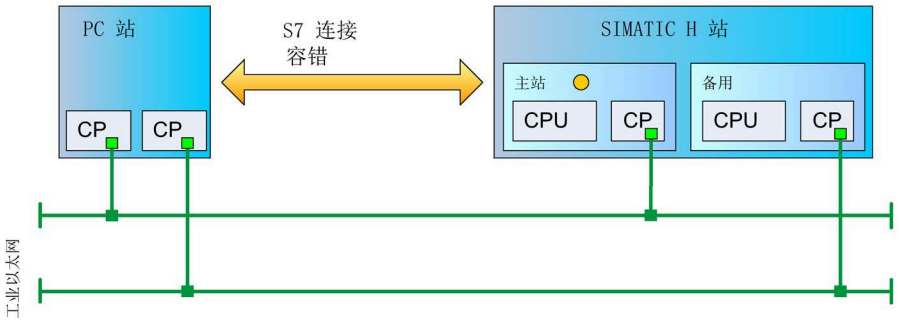

图 2-24 将 PC 站连接到 SIMATIC H 站

本部分介绍了如何使用、组态、调试和诊断容错 S7 连接。

2.7.5.1 容错 S7 连接，概述

PC 站与 S7-400H 自动化系统之间通过冗余通信路径进行通信。通过非冗余连接路径建立标准 S7 连接时，容错 S7 连接可通过这些冗余连接路径实现容错通信。

从应用角度看，容错 S7 连接与标准 S7 连接的作用相同。这意味着上述所有 S7 协议的服务均可使用。而且现有应用无需修改即可使用。

但与标准 S7 连接相比，容错 S7 连接可以同时使用两条连接路径（共四条可选连接路径），这样一来，当其中一条连接路径出现故障时，并不会中止连接。监视和同步机制可确保，主动冗余连接路径失效时会被检测到，并且被动（冗余）连接路径将自动接管通信。连接本身保持已建立的状态。应用程序并不会察觉到该故障切换，但可以通过诊断界面检测到这一过程（另请参见"诊断、调试、维护和运行 (页 67)"部分）

能否在两个 PC 站之间使用容错 S7 连接？

不可以，与标准 S7 连接不同的是，容错 S7 连接只能用于连接 PC 站与 SIMATIC H 站。

容错 S7 连接有哪些变型？

容错 S7 连接的冗余可进行扩展，而且可通过增加 CP 的数量和所用网络的数量来提高冗余度。
可组态以下容错 S7 连接：

- 基于 2 条路径的容错连接
- 基于 4 条路径的容错连接（冗余度增大）

与标准 S7 连接相比，容错 S7 连接具有哪些优缺点？

容错 S7 连接的优点是单个组件的失效可进行弥补。

使用冗余来提高系统可用性通常需要额外的成本和增加所需资源。这同样适用于容错 S7 连接、S7-400 自动化系统组件以及所涉网络组件。

哪些传输协议/媒介可用于容错 S7 连接？

容错 S7 连接只能用于工业以太网网络。可以使用下列传输协议：

- ISO（仅限通过 HARDNET 模块进行的操作）
- ISO-on-TCP（符合 RFC 1006）

虚拟环境中能否同样使用容错 S7 连接？

可以，用户可以在基于 VMware vSphere 平台的虚拟机中使用容错 S7 连接。请确保已阅读“SIMATIC NET PC 软件”安装说明中“使用 VMware vSphere 时的安装和组态”部分的要求、注意事项和限制。

请记住，虚拟机必须具有足够的资源，尤其要始终具备计算能力（使用 ISO-on-TCP 协议时尤其如此）。

哪些模块可用于容错 S7 连接？

所有工业以太网模块均可用于 PC 站中，如下所述：

- 用于 ISO 和 ISO-on-TCP 的 CP 1623（SIMATIC NET PC 软件 V8.1.2 版本以上的 TCP）
- 组态为"IE General"的以太网适配器（SIMATIC NET PC 软件 V8.2 版本以上，仅限 ISO-on-TCP， 个 PC 站上最多 2 个 CP，无法增大冗余度）

有关可使用的特定类型的 CP 数以及可用的连接数的信息，请参见相应的组态限制信息。组态限制信息请参见此条目 ID 下的支持页面：15227599 (https://support.automation.siemens.com/WW/view/zh/15227599)

对于容错 S7 连接，建议仅使用相同类型的 CP。

对于如何使用 S7 H 系统中的接口模块，请阅读相关系统文档。

基于 2 条路径的容错 S7 连接有哪些属性？

此容错 S7 连接具有两条通信路径。一个组件发生故障将导致故障自动切换到另一个冗余通信路径。

以下时序图显示了数据传送期间出现问题时的事件顺序。

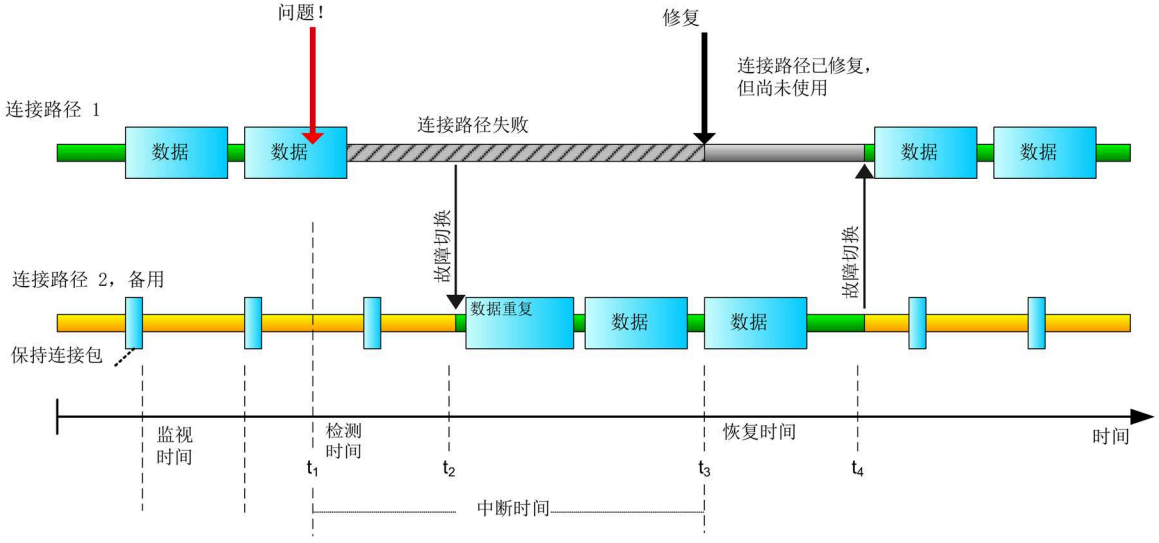

图 2-25　　故障切换到备用路径

容错 S7 连接具有两个可用连接路径，连接路径 1 和备用连接路径 2。最初，路径 1 用于传送用户数据，备用路径则通过周期性传送保持连接包确保其连接状态（该周期比监视时间短）。时间 t_1 处出现问题（红色）。

经过两次监视时间之后（图中显示为检测时间），问题被检测到并且连接路径被标记为不可用。此时立即故障切换到备用连接，继续传送用户数据。未使用的连接路径显示为灰色。如果由于出现问题导致连接路径不可用，则该路径以阴影线显示。

由于监视机制会周期性地检查连接路径，因此只有在经过这样一个周期加以修复后才会故障恢复为路径 1。之后便可继续正常运行。

S7 连接在整个期间均处于建立状态。只有在修复路径 1 之前或修复路径 1 后故障恢复期间（即时间 t_1 和 t_4 之间），备用路径也出现问题时，S7 连接才会中止并需要重新连接。

在此连接路径上存在主动数据通信并且使用 ISO 协议时，检测出线路上中断的时间小于一秒。ISO-on-TCP 的故障切换时间取决于组态的监视时间（见组态）。

基于 2 条路径的容错 S7 连接具有哪些工厂组态？

例如，双路径通信可采用以下组件建立（另请参见下图）：

- SIMATIC H 站，两个机架，每个带一个 CP
- 2 个网络
- PC 站，带 2 个 CP，如 CP 1623

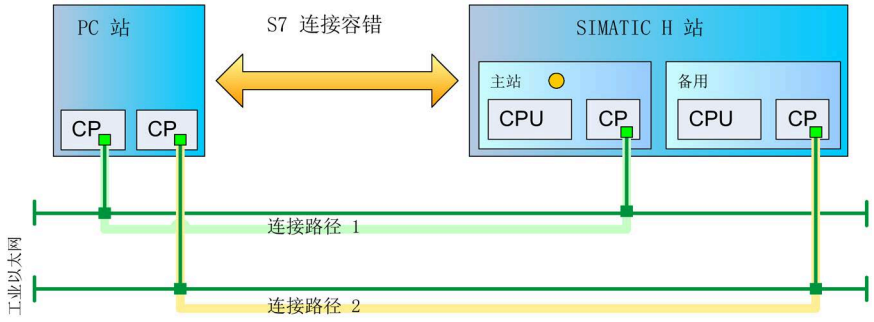

图 2-26　　基于 2 个网络的双路径冗余示例

说明

如果要通过 ISO-on-TCP 将容错 S7 连接用于 H 站的 CPU 417-5H，还可以通过 CPU 的网络接口直接实现此功能。这种情况下，选择 CPU 接口作为连接伙伴（另请参见“组态 (页 65)”部分）。这不适用于使用 ISO 协议的连接。“基于 2 个网络的双路径冗余示例”图显示了如何通过 H 站的 CP 进行通信。

说明

也可以仅仅通过一个网络和 PC 站中的一个 CP 来使用容错 S7 连接。这种情况下，两条连接路径使用相同的网络。如果采取其它适当的措施提高网络基础设施的可用性，则此类组态非常实用。但仍建议使用上述组态。

基于 4 条路径的容错 S7 连接有哪些属性？

与双路径连接相比，基于 4 条路径的容错 S7 连接在出现问题时可使用另外 2 条连接路径。

如果已将容错 S7 连接组态为最大 CP 冗余（4 条路径），有用路径或备用路径发生故障后，则会建立另一个连接路径（假定可用）。根据组态的不同，故障切换过程可能需要花费一些时间。随后连接再次处于"冗余"状态（通过新路径）。

说明

调试期间，务必检查额外的连接路径是否确实可用，例如依次在路径 1 和路径 2 上引入问题，检查系统是否会故障切换到路径 3 和路径 4（另请参见"诊断、调试、维护和运行 (页 67)"部分）。

下图显示了连接路径 1 出现问题时的响应示例：

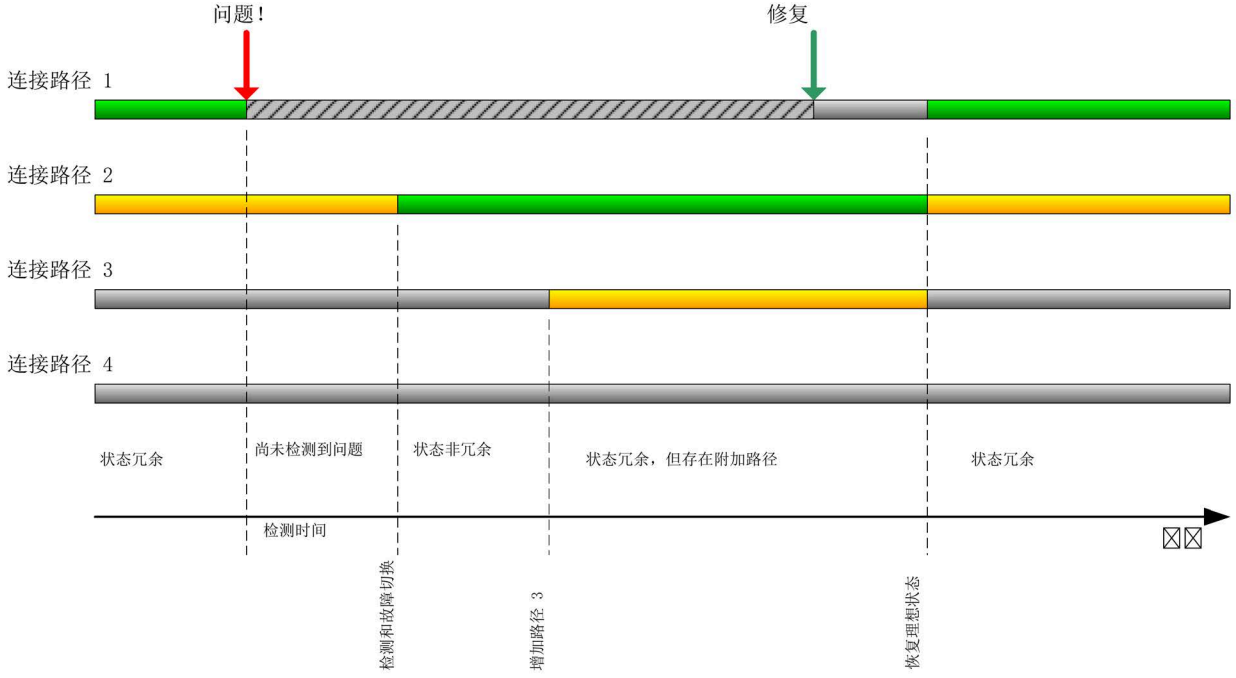

图 2-27　故障切换到冗余度增大的容错 S7 连接的备用路径和路径 3

图中显示了所有 4 条连接路径，其中本例中只使用前 3 条。当出现问题并经过一段检测时间后，路径 2 开始使用（绿色）。检测到路径 1 不再可用时，将添加之前未使用的路径 3 作为备用路径（黄色）。未使用的路径采用灰色显示（由于出现问题而导致不可用的路径以阴影线显示）。

修复之后，检测到路径 1 可再次使用，理想情况下会故障恢复到路径 1。

说明

出现问题并且成功故障切换到附加连接路径后，由于再次存在备用连接路径，因此连接将再次处于"冗余"状态。但这种状态目前并不理想，因为根据组态的不同，可能无法保证组件再次出现故障时仍可继续维持连接。

基于 4 条路径的容错 S7 连接具有哪些工厂组态？

只有 HARDNET 模块可用于 4 路径通信。CPU 接口也不可使用。

基于 2 个网络的 4 路径通信

举例来说，基于四条路径的冗余性更高的容错通信可通过以下组件构建：

- SIMATIC H 站，两个机架，每个带两个 CP
- 2 个网络
- PC 站，带 2 个 CP（如 CP 1623）

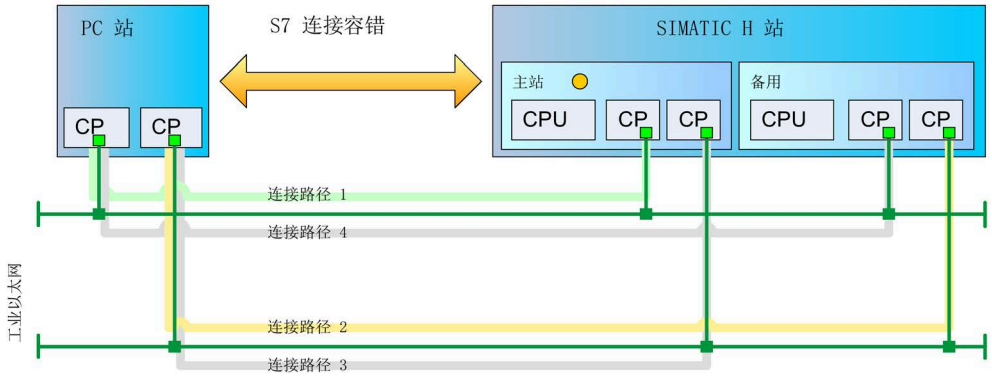

图 2-28　　基于 2 个网络的 4 路径组态示例

说明

CPU 417-5H 的网络接口不适用于 4 路径连接。原因：容错 S7 连接中无法组合运行 CP。但是，除 4 路径连接外，CPU 接口的双路径连接仍可运行。

请注意，当连接路径 2 和路径 3 相继出现故障后，冗余将不再适用，因为路径 1 和路径 3 指向 SIMATIC H 站系统的相同部分，而当系统的该部分出现问题时，将导致连接立即中止。这同样适用于以下示例。

基于 4 个网络的 4 路径通信

另一种方案是下图所示的基于 4 个网络的 4 路径通信。要使用这种方案，需要对 PC 站中的 4 个以太网 CP 全都进行组态。

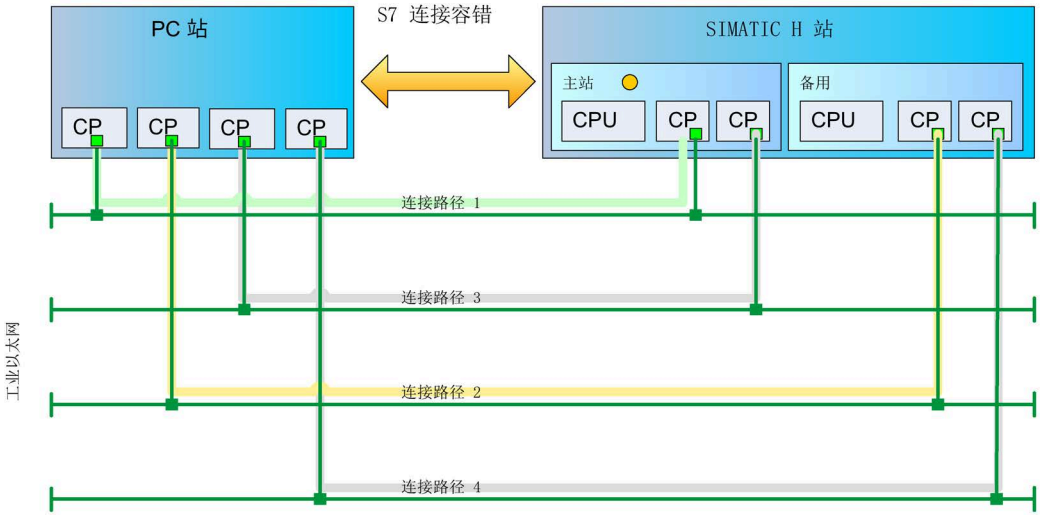

图 2-29　　基于 4 个网络的 4 路径组态示例

说明

最后，还可以实现基于单一网络的 4 路径通信。这种情况下，PC 站中只有 1 个 CP 处于激活状态，而且所有 4 条连接路径均使用同一网络。如果使用其它方法（例如在网络中使用冗余服务器和冗余环网）确保网络和 PC 站的可用性，这种组态非常实用。

使用基于 ISO-on-TCP 的容错 S7 连接时需要注意哪些事项？

使用基于 ISO-on-TCP 的容错 S7 连接时，需记住 PC 站上的连接监视在应用程序的进程空间中运行，并且只有在应用程序进程始终具有足够的系统可用资源（尤其是 CPU 计算能力）时才能够正常工作。

使用基于 SOFTNET 模块和 ISO-on-TCP 的容错 S7 连接时需要注意哪些事项？

操作 SOFTNET 模块和 ISO-on-TCP 协议时，需注意以下事项：

通过 ISO-on-TCP 进行传输最终仍是使用 TCP/IP 协议。借助 SOFTNET，IP 路由由 PC 的操作系统进行处理。这意味着选择地址时需要确保 IP 路由是唯一的，以便各连接路径的通信都基于正确的接口实现。

也就是说：PC 站以及 S7-400H 子站中连接的所有 CP 的 IP 地址必须位于不同的子网中。下图显示了使用 140.1.* 和 140.2.* 这两个子网的示例。确保涉及的所有 CP 上均使用了正确的子网掩码（此处为 255.255.0.0）。这同样适用于 PC 站中安装的所有未组态 SOFTNET 模块。

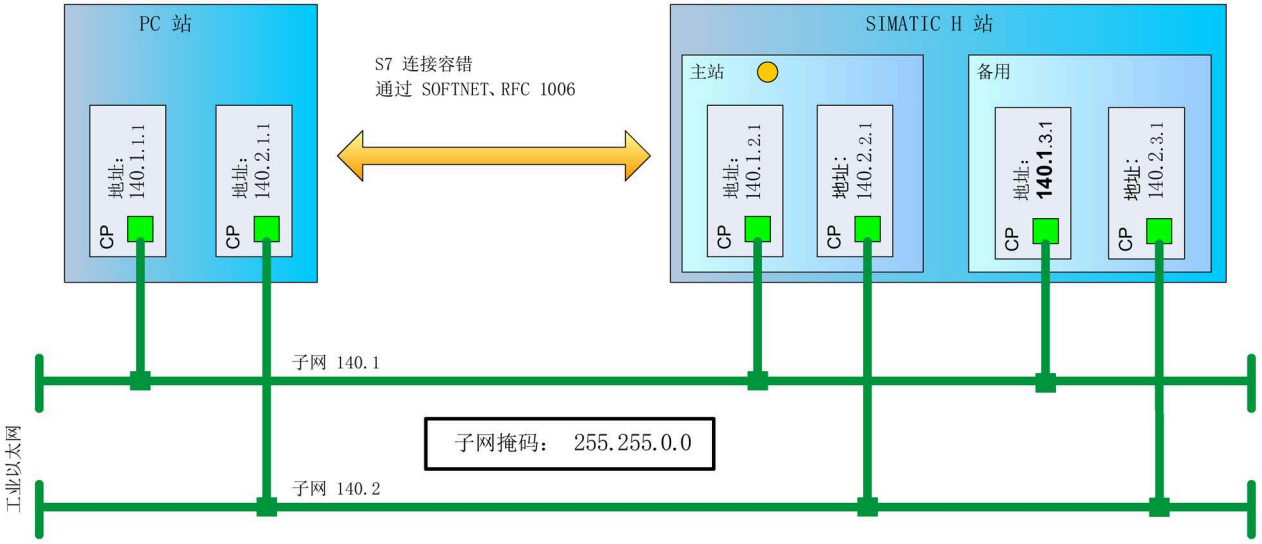

图 2-30　　基于 SOFTNET 和 ISO-on-TCP 的容错 S7 连接

否则将出现以下错误：

- 假设使用的子网已连接（例如通过冗余环网），而且 IP 地址的子网部分不是唯一的。某些情况下，两条连接路径将通过相同的物理连接路径进行路由。只有当其中一条连接路径出现问题时，连接才会建立冗余并中止。
  这类明显的冗余极为危险，因为只有出现问题时才会注意到此点。用户可以并且应该通过逐个断开通信网络中的各个 CP，确定连接是否切换到非冗余状态来对此进行检查（另请参见"诊断、调试、维护和运行 (页 67)"部分）。
- 如果各子网在物理上是独立的，但 IP 地址的子网部分不是唯一的，则当所有连接路径都通过相同网络进行路由时，可能无法建立冗余连接。相应的连接伙伴也不可访问。

说明

使用 HARDNET 模块时，此类问题便不会发生，因为 TCP/IP 通信由模块本身进行处理。但即便如此，用户仍应确保整个网络中的 IP 地址是正确的。

2.7.5.2 组态

使用容错 S7 连接需要执行连接组态。本部分介绍如何组态连接以及需要注意的事项。

组态容错 S7 连接有哪些要求？

要组态连接，需要使用 STEP 7 组态工具。容错 S7 连接只可在 PC 站与 S7-400H 站之间进行组态。

与所有连接组态一样，PC 站必须包含 OPC 服务器和/或应用程序以及所需模块。

现在根据上述其中一种组态来使用连接所需的工业以太网网络对 PC 站与 H 站进行联网。

说明

如上所述，使用 ISO-on-TCP 协议时，应确保涉及的所有网络附件均具有正确的 IP 地址。

版本：

要组态容错 S7 连接，需要 STEP 7 版本 5.1 SP1 或更高版本。

要使用 RFC 1006 协议实现容错 S7 连接，需要 STEP 7 版本 5.5 SP3 或更高版本。对于此协议，以下版本适用于硬件目录中使用的组件：

- OPC 服务器和应用程序：V8.1.2 或更高版本
- "IE General"：V8.1.2 或更高版本

容错 S7 连接如何组态？

该种连接的组态与标准连接的组态类似。确保将"S7 连接容错"指定为连接类型。

组态容错 S7 连接时需要注意哪些问题？

与标准 S7 连接相比，用户需要注意容错 S7 连接的多个附加参数和特性。

连接端点：

建立连接的请求始终由 PC 站发出。这意味着 PC 站始终处于主动状态，而 H 站则作为被动的通信伙伴。因而无法对此进行选择。

故障容错：

在 PC 站与 H 站之间创建连接时，建议您使用两条合适的连接路径。联网之后，也可以选择将冗余增加到 4 条路径。随后便可获得 4 条连接路径的列表。

根据联网方式的不同，您仍可以通过选择其它 CP 接口来改变连接路径列表。

但应牢记：如果使用带工业以太网接口的 H CPU，则无法组态包含有这类接口的 4 路径连接。

协议，监视时间：

与基于工业以太网的标准连接一样，当 ISO 或 TCP 协议 (RFC 1006) 在通信伙伴上均处于激活状态时，您可以决定使用其中哪一个。如果选择 TCP 协议 (RFC 1006)，则 ISO 协议不可用，因为后者未被激活或者不存在，您需要为连接组态输入一个合适的监视时间。

适当的监视时间值取决于容错 S7 连接数，请参见 S7 Redconnect 产品的组态限制信息。使用 TCP/IP 组态容错 S7 连接时，在连接属性中设置这些监视时间。

请注意，乘数是 100 ms，因此，例如输入值 100 时，监视时间为 10 s。将值设置为大于 25 s 是不切实际的，因为 TCP 保持连接监视 (30 s) 经过那么长时间后会导致连接路径终止。

CP 选项：

对于涉及的 S7-400 CP，需启用"快速切换连接"（CP 选项）。

保持连接时间：

确保为所有涉及的 CP 激活 30 s 的保持连接时间（PC 站和 H 站）。此选项为默认设置。

OPC 服务器：

某些情况下，监视时间对于 OPC 服务器的连接建立来说过短。如果 OPC 服务器被组态为仅当需要时才建立连接，则意味着出现问题后，OPC 服务器可能不再建立连接。这种情况下，需延长监视时间来建立连接。

2.7.5.3 诊断、调试、维护和运行

下列选项可用于诊断容错 S7 连接：

- 检测总状态
- 检测各连接路径的状态
- 检测问题
- 检测问题的原因

调试期间，这可以检查所有连接路径是否正常工作，并可人为引入问题，检查连接的响应情况。

执行维护工作后，可使用这些机制来检查系统的状态是否正确并重新建立所有容错 S7 连接。

运行期间，可对连接状态进行监视。影响连接状态的事件的信号会被立即发出。

下列选项可用于诊断容错 S7 连接：

- 使用 S7 连接诊断
- 使用 OPC
- 使用 SAPI-S7 编程接口上的自定义用户程序（请参见“SIMATIC NET PC 软件”DVD 手册集包含的“S7 编程接口”手册中的“容错连接的诊断服务”部分）

什么是 S7 连接诊断？

S7 连接诊断属于“SIMATIC NET PC 软件”的应用程序，能够以列表视图的形式显示所有已组态的 S7 连接及其状态和其它信息。

状态	应用程序名称	连接名称	接口	远程地址	类型	产品 ID	待机 ID
确定 冗余	Application	S7 appl_conn_1	-----	-----	H	1	2
确定 冗余不佳	Application	S7 appl_conn_2	-----	-----	H	1	3
确定 冗余不佳	Application	S7 appl_conn_3	-----	-----	H	1	3
确定 冗余	Application	S7 appl_conn_4	-----	-----	H	1	2
确定 冗余	Application	S7 appl_conn_5	-----	-----	H	1	2
确定 冗余	OPC Server	S7 opc_conn_1	-----	-----	H	1	2
确定 冗余不佳	OPC Server	S7 opc_conn_2	-----	-----	H	1	3
确定 未冗余	OPC Server	S7 opc_conn_3	-----	-----	H	1	-
确定 未冗余	OPC Server	S7 opc_conn_4	-----	-----	H	1	-
确定 冗余	OPC Server	S7 opc_conn_5	-----	-----	H	1	2
确定	OPC Server	S7 opc_conn_6	CP1623.RFC1006.2	140.102.0.6	S	-	-
未使用	OPC Server	S7 opc_conn_7	CP1623.RFC1006.2	140.102.1.6	S	-	-

图 2-31　　S7 连接诊断

当影响连接状态的问题发生时，连接诊断会立即反映此问题。图中显示了一些已组态 S7 连接的示例。可分别通过类型“H”和类型“S”来识别容错 S7 连接和标准连接。

所建立的状态为“正常冗余”的标准连接和容错 S7 连接以绿色符号指示。

“S7 连接诊断”图中显示了当多个连接所使用的其中一条连接路径出现问题时的情况。结果一目了然：使用该条路径的标准连接被中止。双路径连接可继续运行，但不再是冗余状态。如果另一条连接路径再发生故障，将会导致连接中止。4 路径连接加入了更多的路径，并且只有当目前正在使用的两条路径中的其它组件出现问题时才会导致连接中止。

执行修复后，所有的容错 S7 连接均会自动恢复。但标准连接则需要重新建立。

S7 连接诊断的更多属性

连接诊断也可监视已组态的标准 S7 连接。

双击连接或在选择连接后按下 Enter 键，将显示相关连接的更多详细信息。

组态发生更改后，连接列表的显示会自动刷新。

该应用程序的帮助系统介绍了更多的操作员输入。

如何使用 OPC 诊断容错 S7 连接？

就像标准 S7 连接那样，您可以使用 SIMATIC NET OPC 服务器提供的 S7 特定诊断变量来诊断容错 S7 连接。与标准连接相比，容错 S7 连接具有扩展的状态和连接路径状态。可以使用 OPC 客户端应用程序读取和监视这些变量。

2.7.6 S7 协议 - 如何组态？

S7 协议的组态方式如下

必须先对连接进行组态，而后才可进行基于 S7 协议的通信。为此，提供了“SIMATIC STEP 7 Professional”组态工具。已组态的连接将通过在组态过程中指定的唯一连接名称进行标识。S7 协议预定义了两个连接类型：

- S7 连接：基于 PROFIBUS 或以太网的连接
- 容错 S7 连接：基于冗余连接路径的连接

每个已组态的连接都必须进行参数设置。创建连接后，组态工具将为这些参数设置默认值，用户可不作任何修改即采用这些值。基本参数包括：

- 传输层的服务访问点（TSAP）。
- 连接类型：
  - 基于 PROFIBUS 的 S7 连接
  - 采用 TCP/IP 协议并基于以太网的 S7 连接
  - 采用 ISO 传输协议并基于以太网的 S7 连接

什么是未组态的 S7 连接？

通常情况下，与伙伴设备的连接都在组态中指定。为此，提供了“SIMATIC STEP 7 Professional”组态工具。但是也存在例如必须由伙伴设备读取数据的应用，或者必须写入或监视变量的应用。实施此任务可以不通过组态，甚至是第三方软件也无需费力即可地访问通信变量。

使用未经组态的 S7 连接有哪些要求？

要允许访问未经组态的设备，必须已知相应伙伴设备的所有与通信相关的数据。其中包括连接名称、访问点（CP 选择）、远程 TSAP、站点地址和其它数据。

如何创建未组态 S7 连接？

未组态 S7 连接可通过 OPC 服务器或通过“COMLS7”的“通信设置”组态工具进行创建。

未经组态的 S7 连接有哪些优缺点？

使用未经组态的 S7 连接的好处是，可更快地访问伙伴设备。

使用未经组态的 S7 连接的缺点是，必须已知伙伴设备的所有通信相关信息。除此之外，未经组态的连接还不支持任何缓冲区发送/接收服务。

2.7.7 S7 协议 - 有哪些优缺点？

S7 协议的优点

使用 S7 协议具有以下优点：

- 所有服务均可基于 PROFIBUS 和以太网无限制被使用。
- 无需编程伙伴便可访问伙伴设备。
- 使用密码进行访问控制。
- 访问（读取/写入/删除）自动化系统的可加载区域。
- 报警将进行缓冲并且不会丢失。
- 面向连接的协议的所有优点

S7 协议的缺点

使用 S7 协议具有以下缺点：

- 取决于供应商，S7 协议仅可在 SIMATIC S7 范围内实施。
- 与 S5 通信不兼容。

2.8 SNMP 协议

2.8.1 SNMP 协议 - 它是什么？

SNMP 协议

SNMP 协议 （Simple Network Management Protocol，简单网络管理协议）是一个用于管理网络的基于 UDP 的开放式协议。 其允许对很多网络组件进行集中网络管理，例如路由器、网桥、集线器、打印机、服务器和工作站。 SNMP 的主旨在于降低管理功能的复杂性，使不同网络组件之间的数据或信息交换更加透明。 SNMP 协议支持监视、控制和管理任意的兼容 SNMP 的网络组件。

2.8.2 SNMP 协议 - 典型系统组态的外观如何？

以下部分介绍了不同设备之间借以实施 SNMP 协议数据通信的以太网典型系统组态的外观。

SNMP 协议的系统组态示例

对于以太网 SNMP 协议通信，SIMATIC NET 范围仅包含适用于 PC 和工作站的通信模块。这种环境下可以使用 CP 1623 或其它供应商提供的模块等通信模块。SIMATIC NET 范围内的其它 SNMP 兼容模块为交换机，例如 SCALANCE X300 或 SCALANCE X400。该组态可通过任意的 SNMP 兼容网络组件进行扩展，其中包括其它供应商的组件。

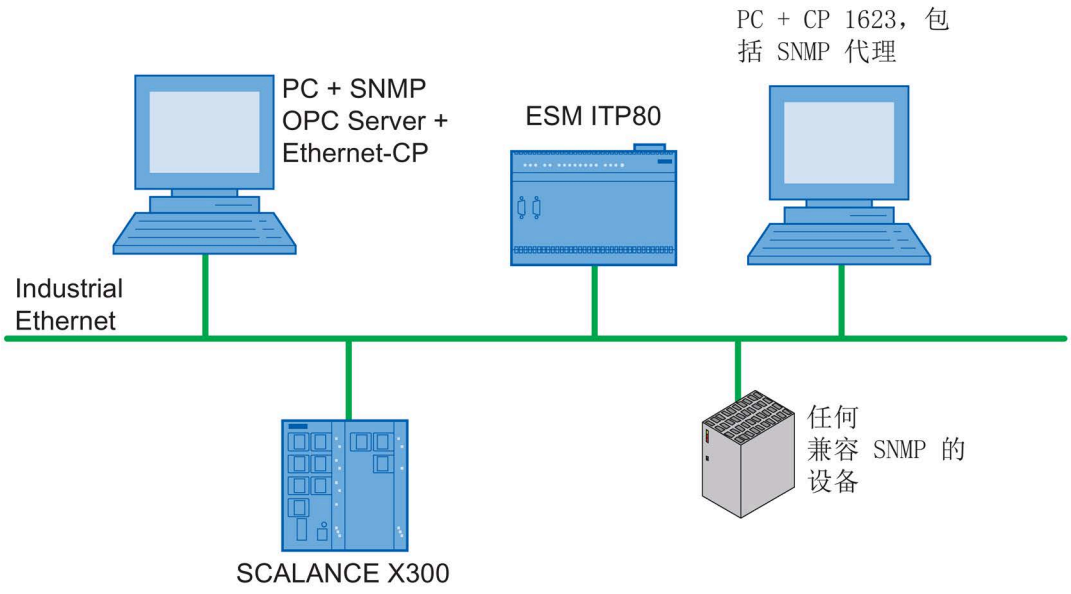

图 2-32　　以太网的典型系统组态

2.8.3　　SNMP 协议 - 工作原理是什么？

SNMP 协议工作原理

SNMP 协议根据客户端-服务器模型来工作。 SNMP 代理在受管理的网络组件上充当服务器，管理可用数据，并控制网络组件。 SNMP 管理器充当客户端，并可通过不同 SNMP 服务与 SNMP 代理交换数据、监视 SNMP 代理并甚至组态该代理。

SNMP 协议如何访问数据？

SNMP 代理管理 MIB（Management Information Base，管理信息库）中可用的数据。MIB 是一种表格类型，其中的所有数据都按一种结构化的形式进行存储。 SNMP 管理器可通过 SNMP 服务读出代理的 MIB，并从而可访问 SNMP 管理器中所需的具体数据，或者访问需要在 SNMP 代理上覆盖的数据。

2.8.4 SNMP 协议 - 提供哪些通信服务？

SNMP 协议提供以下通信服务

对于 SNMP 管理器和 SNMP 代理之间的通信，SNMP 协议基本上提供五种通信服务。

- **获取请求**： 通过"获取请求"服务，SNMP 管理器可向 SNMP 代理请求该代理在其 MIB 中管理的数据。
- **获取下一请求**： "获取下一请求"服务允许 SNMP 管理器访问 SNMP 代理上的下一个数据。
- **获取响应**： "获取应答"服务是"获取请求"或"获取下一请求"的应答，其始终由 SNMP 代理向 SNMP 管理器发送。
- **设置请求**： 要向 SNMP 代理中写入数据，SNMP 管理器可使用"设置请求"服务。
- **TRAP**： 特殊数据可主动地或由事件驱动地从 SNMP 代理发送到 SNMP 管理器。SNMP 代理然后将使用"陷阱"(trap) 服务。

2.8.5 SNMP 协议 - 如何组态？

SNMP 协议的组态方式如下

必须先组态所有 SNMP 兼容伙伴设备组态，才可进行 SNMP 协议通信。 为此，提供了"SIMATIC STEP 7 Professional"组态工具。 SNMP 兼容伙伴设备的多个参数通过 HW Config 用户程序指定，这些参数可对该设备进行唯一标识。 这些要组态的参数包括：

- 设备的名称： 唯一且在技术上相关的名称
- 设备的地址： 以太网 IP 地址
- 设备规约：描述可通过 SNMP 协议提供的设备信息的结构

2.8.6 SNMP 协议 - 有哪些优缺点？

SNMP 协议的优点

SNMP 协议具有下列优点：

- 许多供应商都支持的开放式协议。
- 在以太网网络中应用广泛。
- 可支持大量不同的网络组件，例如交换机、打印机、PC、网络适配器。
- 通信可以采取事件驱动式，进而降低网络负载。

SNMP 协议的缺点

SNMP 协议具有下列缺点：

- 没有诊断协议，因而无法进行任何网络诊断。
- 无任何统计可用。
- 无法进行任何参数分配。

2.9 PROFINET IO 通信

2.9.1 PROFINET IO - 它是什么？

这是 PROFINET IO

PROFINET IO 是在工业以太网上实施模块化和分布式应用的一项自动化概念。 使用 PROFINET IO，分布式 I/O 和现场设备可集成到以太网通信中。 其采用 PROFIBUS DP 的标准 IO 视图，即现场设备的非时间关键用户数据将按一定的周期传送，或时间关键数据在实时通道内传送到自动化系统的过程映像。

PROFINET IO 描述了一个建立在 PROFIBUS DP 基础上的设备模型，且其基于插槽和通道（子插槽）。 PROFINET IO 的工程组态也与 PROFIBUS DP 中的相同，因而也为系统集成商所熟知。 分布式现场设备通过组态分配给可编程控制器，这些设备也称为 PROFINET 设备。

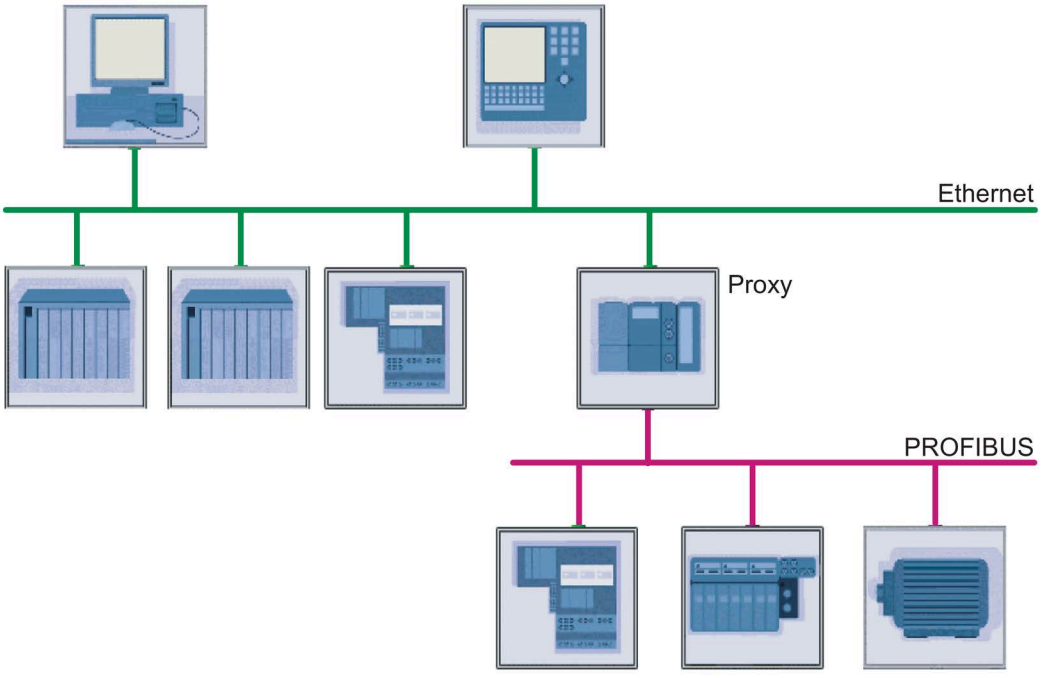

图 2-33 在以太网通信中集成 PROFIBUS

2.9.2 PROFINET IO - 典型系统组态的外观如何？

PROFINET IO 工厂组态示例

以下部分将介绍 PROFINET IO 的典型系统组态的外观。 此部分将明确 PROFINET IO 所提供的各个选项及其所具备的灵活性。

这里将以一个工业以太网为例，其中连有类型为 IO 控制器和 IO 设备的 PROFINET 设备。同时也会将一个 PROFIBUS 网段作为总线子系统通过 IE/PB Link 集成到该网络中。

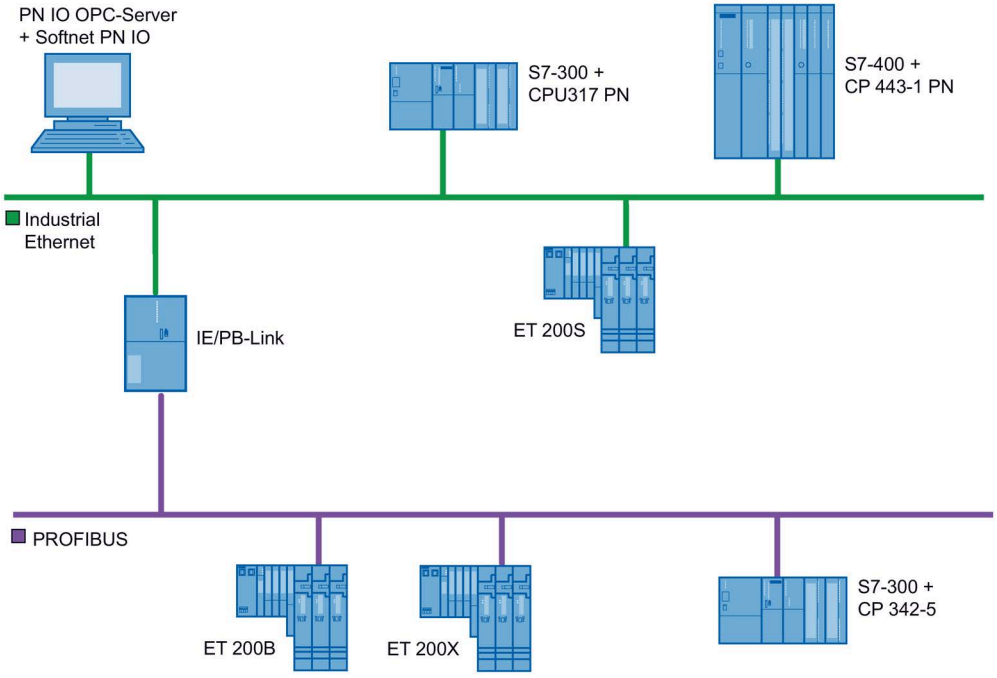

图 2-34 PROFINET IO 的典型系统组态

表格 2- 3 各具体 PROFINET 设备的说明

PROFINET 设备	设备类型	说明
1	IO 控制器	PC 中的 CP（例如以太网 CP 1616）是 PROFINET IO 控制器，并与多个 IO 设备通信。 例如，该 PC 上运行 PN IO OPC 服务器或 PN IO 应用程序。
2	IO 控制器	S7-300 设备充当 IO 控制器，并与多个 IO 设备通信。
3	IO 控制器	S7-400 设备充当 IO 控制器，并与多个 IO 设备通信。

PROFINET 设备	设备类型	说明
4	IO 设备	从 PROFINET IO 的角度上说，IE/PB Link 具有代理功能，其作为以太网中的一个 PROFINET IO 设备而明显代表各个底层 PROFIBUS 设备。
5	IO 设备	ET 200S 设备充当 IO 设备，并被分配给一个 IO 控制器。

2.9.3 PROFINET IO - 工作原理是什么？

PROFINET IO 的工作原理如下

通过 PROFINET IO，将分布式 IO 集成到工业以太网中。控制器和设备根据**提供方-使用方模型**工作，即提供方生成并发送数据，而使用方接收该数据并进行处理。控制器-设备原则可以与我们了解的 PROFIBUS DP 中的主站-从站原则进行比较。

从通信角度看，以太网中的所有 PROFINET 设备都具有相同的权限。仅基于该组态，为设备分配类型（根据提供方-使用方模型指定通信类型）。

在 PROFINET IO 中，区分以下三种设备类型：

- **IO 控制器**
  IO 控制器是运行自动化程序的可编程控制器或者 PC 中的 CP，例如用于实施 OPC 服务器的 CP。
- **IO 设备**
  IO 设备是分配给 IO 控制器的分布式现场设备。
- **IO 管理器**
  IO 管理器是具有调试和诊断功能的 PC/PG。

可以通过以下通道，在 IO 控制器和 IO 设备之间传送数据：

- 周期性用户数据，基于实时通道
- 事件驱动式中断，基于实时通道
- 非周期性读取和写入数据记录、参数分配和组态以及读取诊断信息，基于 UDP/IP 的标准通道（NRT 通道）

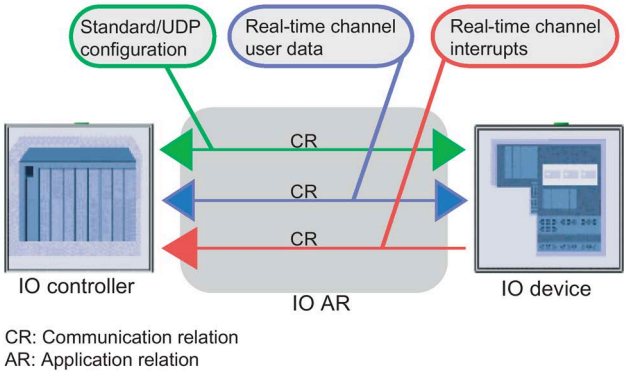

图 2-35　　IO 控制器和 IO 设备之间的通信原理

在 IO 控制器和 IO 设备之间开始通信时，UDP/IP 通道上将建立一个应用关系。根据上述用于传送组态数据、用户数据和中断的通道，这可包含多个通信关系。

同时也会为 IO 控制器和 IO 管理器之间的通信建立一个应用关系。这里将使用 UDP/IP 通道来传送诊断数据和进行上传和下载功能。

从 IO 管理器到 IO 设备的通信也基于应用关系框架内的 UDP/IP 通道进行。除诊断数据外，还将传送状态信息和参数数据。

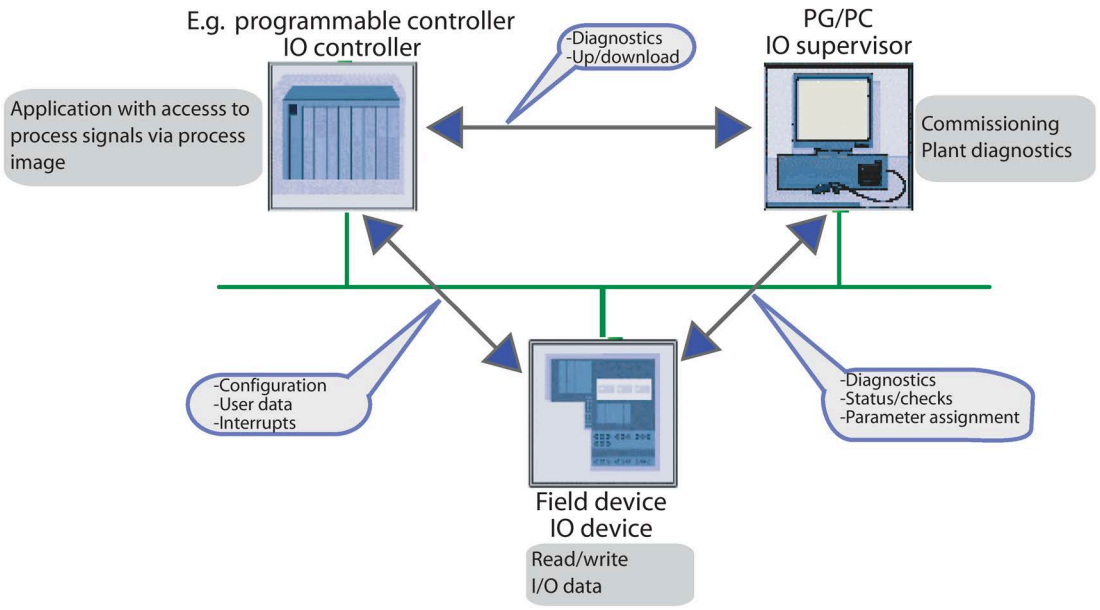

图 2-36　　PROFINET IO 的功能范围

PROFINET IO 中使用的协议

对于 PROFINET IO，在 PROFINET IO 设备间的通信开始时，将使用 UDP/IP 来进行数据交换，为分布式现场设备和 IO 设备分配参数并进行诊断。将 RPC 协议用作应用协议。RPC 协议（远程过程调用）是可支持在网络中实施分布式应用的一项协议。而且凭借该协议，HMI 站或工程组态系统还可作为 PROFINET IO 设备的 IO 管理器而接入。要传送用户数据和中断，则将使用 PROFINET 实时通道。

在典型 PROFINET IO 组态中，存在一个 IO 控制器，由它来通过通信关系与多个分布式现场设备（即 IO 设备）进行周期性数据交换。在每个周期中，输入数据将由指定的现场设备发送给 IO 控制器，输出数据将按相反方向被发送到相关现场设备。对通信关系的监视通过监视周期性数据的抵达情况来实现。如果周期性预期信息未抵达，则 IO 控制器会将其识别为相应的 IO 设备已发生故障。

2.9.4 采用等时实时通信 (IRT) 的 PROFINET IO

PROFINET IO 的三个性能等级

与基于 TCP/UDP 和 IP 的通信相比，RT 通信由于省略了多个协议层（ISO/OSI 参考模型的第 4 - 6 层）而具有更短的更新时间。

通过使用 VLAN 优先级，RT 通信还按照较高的优先级来传送实时帧。 实时帧的优先级比 TCP/UDP 数据的高，加上交换机上的存储时间更短，这些都进一步地加快了传输速度。 从而，原本更新时间便低于 10 ms 的 PROFINET IO 实时通信功能又提高了一个层次，可满足运动控制应用的极高要求。

等时实时通信 (IRT) 专为运动控制领域而设。 对于 IRT 通信，由于在帧的优先处理和更短的交换机存储时间之外又实施了更多功能，从而可实现 1 ms 以下的更新时间。 下文介绍了这些条件。

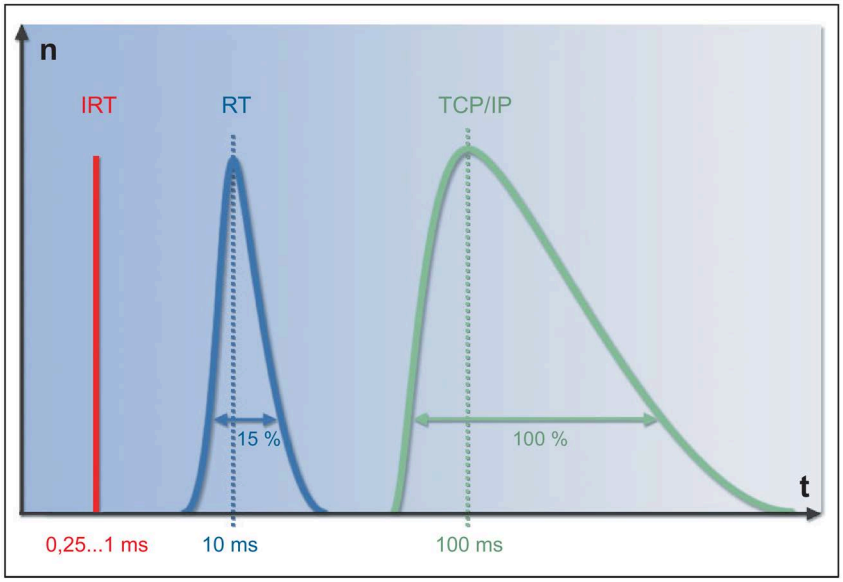

图 2-37 IRT、RT 和 TCP/IP 通信的更新时间对比

什么是等时实时通信？

IRT 的高性能主要通过以下三个特性实现：

- 将传送周期划分为两个时间间隔
- 通过使各个节点同步实现等时传输
- 对通信进行时间和路径方面的规划

IRT 传送周期的两个时间间隔

要在固定的时隙中实现实时帧的优先传送，传送周期划分为两个时间间隔：

- 一个面向实时帧的确定性 IRT 通道
- 一个面向 TCP/UDP 和 RT 通信的开放性非时间关键通道

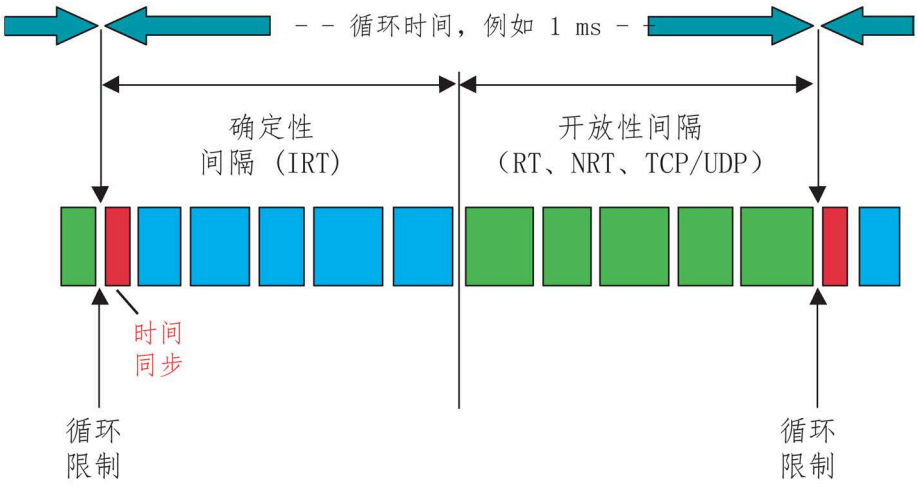

图 2-38　　使用 IRT 通道时传送周期的结构

实时 IRT 帧被分配到确定的时间间隔中，并且该时段仅限于传送此类数据。

IRT 的时间同步

IRT 通信的各个周期在时间上保持同步，可允许等时传送各个 IRT 帧，从而极大程度地缩短了周期时间。

为使 IRT 同步域中所涉及的各个节点实现时间同步，特别组态了一个用于发布同步帧的同步主站。 按照同步主站的时基进行同步的设备称为同步从站。 同步主站和同步从站共同形成 IRT 同步域。 与正常的域相比，IRT 同步域仅包含采用 IRT 的 PROFINET 设备。

用户程序如何访问等时过程数据？

用户程序在开放性间隔时间内访问过程数据，而为 IRT 通信而组态的数据将在下一个确定的间隔时间内通过 IRT 通道传送。这样便可实现一致的数据传送。

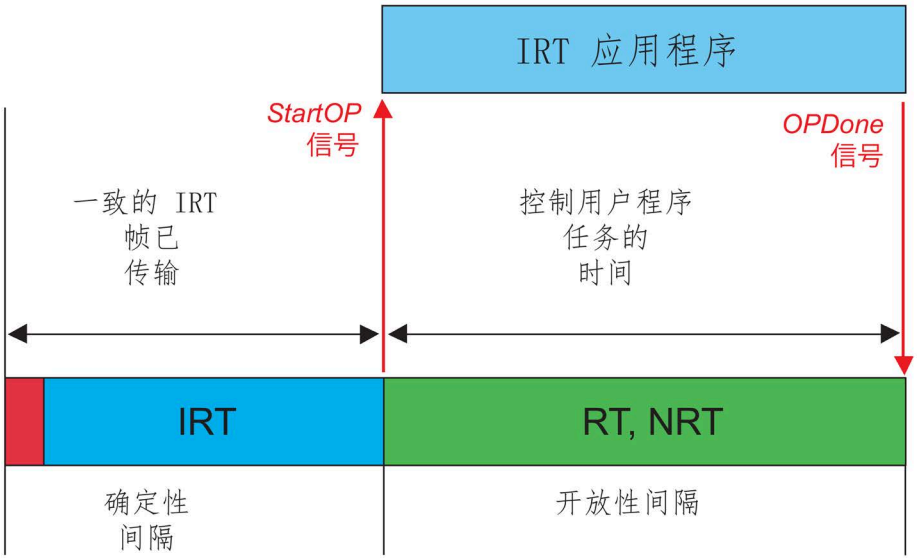

图 2-39　　IRT 通信中传送周期的时间同步

在 IRT 帧发送后，在 IRT 时间间隔结束时用户界面将输出一条“StartOP”消息（开始运行）。

在“StartOP”消息后，用户程序可在开放性时间间隔内执行其周期性控制任务。 在此开放性时间间隔内，非时间关键的 NRT 帧和 RT 帧可通过开放性通道传送。

在程序周期结束时，用户程序将输出“OPDone”确认信息。 只有在“OPDone”确认发出信号后，新周期才可开始。

确定性时间间隔的持续时间在组态过程中指定。

对 IRT 通信进行时间和路径方面的规划

IRT 通信通过以下 Siemens 组态工具设置：

通过规划各个伙伴间的通信路径，可实施最短的传送周期。 为实现此目的，该路径中各个 IO 设备和 IRT 交换机之间的连接应在一个拓扑规划阶段中进行组态，而且组态时必须将电缆长度考虑在内。 可实现的最短周期通过组态工具进行计算，具体结果取决于节点之间的传送时间。

IRT 通信的硬件要求

为使等时数据传输的周期时间小于 1 ms，且连续两个周期之间的抖动在 1 µs 范围，在控制器和设备上以及它们之间的交换机上都需要专用的 IRT ASIC。

Siemens 提供了多种基于以太网 ASIC ERTEC 200 和 ERTEC 400 且支持高性能 IRT 通信的组件：

- 配有 IO-Base 软件的 CP 1604 和 CP 1616 通信处理器
- SCALANCE X204IRT 和 X202IRT 交换机
- 更多组件正在开发中。

使用 SOFTNET PNIO 实现 IRT 的注意事项

说明

SOFTNET PNIO 不支持 IRT。

2.9.5 PROFINET IO - 提供哪些通信服务？

PROFINET IO 提供以下通信服务

PROFINET IO 提供多种面向 PROFINET IO 控制器与 PROFINET IO 设备间通信的通信服务。对初始化服务和生产服务加以区别。 初始化服务包括：

- **IO 控制器状态**： 凭借"IO 控制器状态"服务，IO 控制器可查询并更改其自身的状态。IO 设备共定义了三种状态 CLEAR、OPERATE 和 OFFLINE。
- **激活 IO 设备**： "激活 IO 设备"服务允许 IO 控制器将 IO 设备设置为活动状态。
- **取消激活 IO 设备**： "取消激活 IO 设备"服务允许 IO 控制器将 IO 设备设置为未激活状态。

生产服务包括：

- **读取 IO 数据**： 凭借"读取 IO 数据"服务，IO 控制器可读取 IO 设备的周期性输入数据。 同时，远程状态信息（提供者状态）也从 IO 设备读取，并且本地状态信息（消费者状态）也将传送到 IO 设备中。
- **写入 IO 数据**： 通过"写入 IO 数据"服务，IO 控制器可修改周期性发送到 IO 设备的输出数据。 同时，本地状态信息（提供者状态）也将传送到 IO 设备中。
- **接收和应答中断**： 凭借"接收和应答中断"服务，IO 控制器可接收 IO 设备的信息，并可对此向 IO 设备进行应答。
- **读取/写入数据记录：** 凭借此服务，IO 控制器可与 IO 设备进行非周期性通信。 IO 控制器从 IO 设备读取数据记录或向 IO 设备中写入数据记录。

2.9.6 PROFINET IO - 如何组态？

PROFINET IO 的组态方式如下

要允许 PROFINET IO 通信，每个 PROFINET IO 设备都必须进行组态。 为此，提供了 "SIMATIC STEP 7 Professional"组态工具。

每个 PROFINET IO 设备都必须进行参数设置。 当创建设备后，组态工具将为这些参数设置默认值，即在用户未修改情况下可以采用的值。 基本参数包括：

- 更新时间
- 设备的访问地址

IRT 通信的组态注意事项

对基于时间的 IRT 通信进行组态时，不仅必须组态通过 IRT 通道进行通信的节点，还应指定位于它们之间的交换机的地址。 当交换机启动时，控制器会自动向它们传送规划数据，以允许创建传送列表。

IRT 同步域中采用 IRT 的 PROFINET 设备可属于一个或多个 IO 系统。 但重要的是，在一个 IO 系统内所有采用 IRT 的 PROFINET 设备都仅属于一个 IRT 同步域。 IRT 同步域不可重叠。 与不同 IRT 同步域的 PROFINET 设备连接仅限在不支持 IRT 的 PROFINET 设备或端口上进行。

IRT 同步域只能包含支持 IRT 的交换机，不可包含标准交换机。

将单独的数据包分配到开放性 NRT 通道或分配到 RT 或 IRT 通道也需要使用上述组态工具。

2.9.7 PROFINET IO - 有哪些优点？

PROFINET IO 的优点如下

使用 PROFINET IO 具有以下优点：

- 投资保护
- 系统扩展简单直观
- 最小化安装、工程设计和调试的成本
- 通过集成 PROFIBUS 可纵向集成自动化金字塔上的各个级别
- PROFIBUS 的组态限制已扩展，同时也实现了更高的性能
- 在一条电缆上同时进行实时通信和基于 TCP 的通信
- 实时通信可从强大到高性能和等时的范围内进行扩展
- PROFINET 设备间的标准化通信

2.10 SIMATIC NET 安全性

有关 SIMATIC NET 安全性的所有信息，请参见手册《工业以太网安全 – 设置安全性》。该文档位于“SIMATIC NET PC 软件”“doc”文件夹中的手册集上，以及采用以下条目 ID 的支持页面上：

60166939 (http://support.automation.siemens.com/WW/view/zh/60166939)

OPC 接口基本知识

3

概述

下面部分将总体介绍 OPC 的基本知识和 SIMATIC NET 中的 OPC 应用。

本章将解答 OPC 基本知识方面的问题，可帮助您熟悉 OPC 这一术语，并为各具体术语提供解释。其中简要概述了将 OPC 与 SIMATIC NET 结合使用时的优势，并深入介绍通过变量访问过程数据（OPC 数据访问）、传输硬件中断和事件（报警和事件）以及从归档读取历史数据 (OPC HA) 的规范。

同时还将介绍 OPC 统一架构 (OPC UA) 规范的属性。最后，同样重要的是，您还可以获得有关 OPC 性能的信息。

在您阅读完本章的内容以及本手册第 2 卷中有关 OPC 使用的详细信息后，您就不会遇到什么困难。

3.1 OPC 简介

3.1.1 OPC - 是什么？

OPC 是与供应商无关的软件接口，通过它可在不同供应商的软件和硬件间进行数据交换。

OPC 有何用途？

简言之，“OPC 什么都能干”。

在 OPC 之前，通过软件应用程序来控制不同供应商的硬件是一件非常费力的事。这是因为存在着大量不同的系统和协议。而且对于每个供应商和每个协议，用户都必须使用专门的软件来访问特定接口和驱动程序。这意味着用户程序取决于供应商、协议或系统。在 COM 或 DCOM 基础上的 OPC，作为一个统一且与供应商无关的软件接口，为自动化工程设计中的数据交换带来了革命性的变化。

以下示意图总体地介绍了 OPC 的性能和灵活性。对于其中的各个组件，您将在本手册的其余部分看到，而且它们的解释也会同时给出。

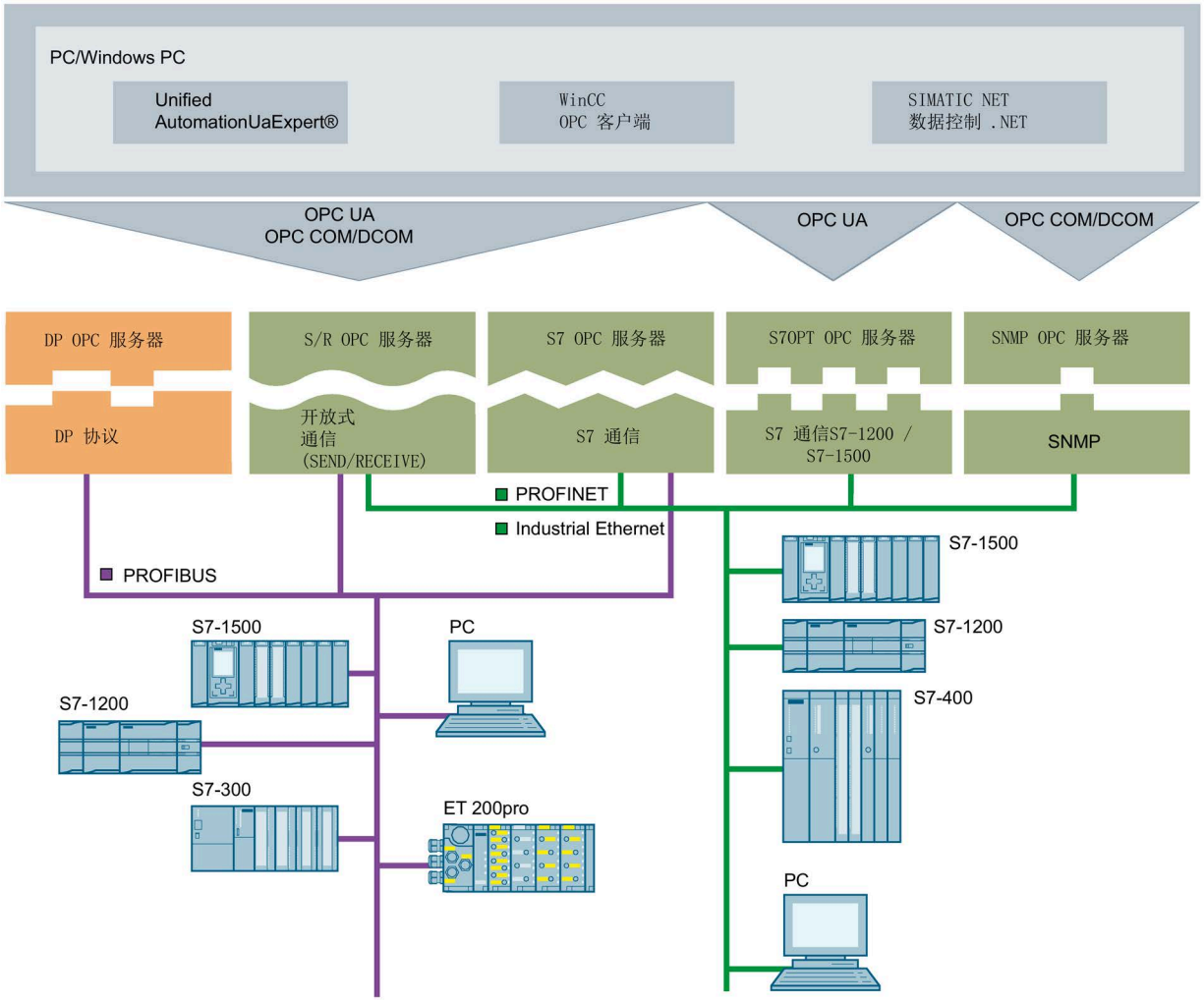

图 3-1　　与 OPC 服务器的系统集成

3.1.2 OPC 接口 - 有何作用？

OPC 接口属于在 PC 上运行且充当操作员控制平台和系统/其它应用程序的监视平台的软件。它位于特定应用程序下。

作为一项工业标准，OPC 定义了工业环境下不同应用领域的信息交换。

OPC 接口基于什么原理？

OPC 接口的应用基于客户端-服务器模型。其中一个组件作为服务器，通过接口为其它组件提供服务。另一个组件作为客户端使用这些服务。应用程序可确定系统中设置了哪些 OPC 服务器。其可对这些服务器中的一个或多个进行寻址，并检查该服务器所提供的服务。由于多个不同的 OPC 客户端可同时访问同一台 OPC 服务器，所以同一数据源可供所有 OPC 兼容应用程序使用。

提供过程数据的模块（通信系统、测量设备等）制造商提供针对其模块的 OPC 服务器，该服务器负责连接特定数据源。

OPC 接管这些任务

您可通过 OPC 接口监视、调用和处理 PC 的系统数据和事件。

OPC 接口包括哪些？

OPC 基金会自 1996 年以来一直在建立 OPC 接口规范。目前，自动化工程设计方面具有以下规范：

- 针对基于过程变量的数据交换：数据访问
- 针对报警和事件处理：报警和事件
- 针对通过 Internet 进行的数据交换：数据访问 XML
- 针对 OPC 服务器间的横向数据交换：数据交换
- 针对配方作业：批生产
- 针对访问归档数据：历史数据访问
- 针对集成众多 OPC 规范：OPC 统一架构

作为工业通信系统的接口，SIMATIC NET OPC 服务器可提供数据访问、报警和事件和统一架构这几项功能。

3.1.3 OPC 服务器 - 它是什么？

OPC 接口所基于的原理是，初始化过程（发送请求，发布作业）与响应过程（处理请求和作业）之间的协作，即客户端和服务器两个方面的协作。

OPC 服务器

提供数据的 OPC 组件称为 OPC 服务器。 它们可为现有的各通信系统提供连接手段。 除服务之外，它们还可为 OPC 客户端提供不同数据源的信息；这些数据源可以是硬件驱动的数据源或软件组件。 数据需要来自诸如接口、现场总线卡、测量设备或控制器等。 每个 OPC 服务器都有一个唯一的名称进行标识。

服务器名称从何而来 (ProgID)？

每个 OPC 服务器都有一个由供应商指定的唯一名称来进行标识。 根据 COM 标准，这些名称称为 ProgID。 通过 ProgID，可具体地寻址各单独 OPC 服务器。

有哪些服务器类型？

有三种类型的 OPC 服务器。 根据在通信系统中的集成方式，它们称为：

- 本地服务器（进程外服务器）
  （此服务器位于本地计算机中）
- 远程服务器（进程外服务器）
  （此服务器位于网络中的其它计算机内）
- 同进程服务器
  （此服务器的性能更强）

OPC 服务器的供应商会指定服务器是同进程服务器还是本地服务器。 是否作为远程服务器则由用户进行组态。

对于三种服务器类型，方法调用的语法都相同。

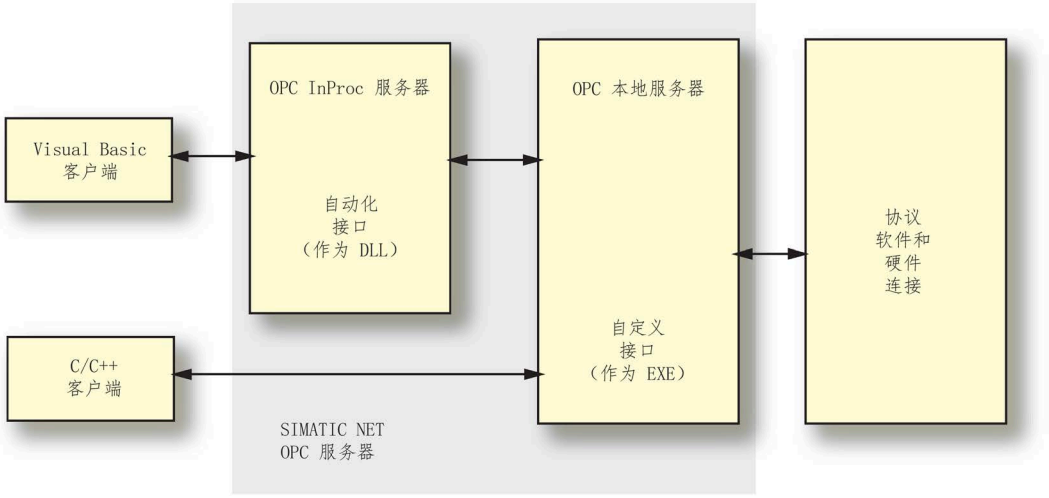

图 3-2　　OPC 服务器

3.1.4　　OPC 客户端 - 它是什么？

OPC 接口所基于的原理是，初始化过程（发送请求，发布作业）与响应过程（处理请求和作业）之间的协作，即客户端和服务器两个方面的协作。

这是 OPC 客户端

以 OPC 服务器为数据源的 OPC 组件称为 OPC 客户端。

是否可购买 OPC 客户端？

OPC 客户端通过标准软件形式提供。 同时也提供可组合使用的软件模块，以便适合您的特定用途并构建一个有效的客户端。

能否创建用户自己的 OPC 客户端？

为理想地满足您系统的具体要求并尽可能地实现最佳的性能，您可使用多种编程语言创建自己的 OPC 客户端（例如 Visual Basic、C、C++ 和 C#）。

有哪些属性必须考虑？

部分 OPC 服务器属性（例如变量名）在 OPC 标准中并未定义，但取决于自动化系统或设备的属性（举例来说），所以由供应商指定。 为使 OPC 客户端能无故障地配合不同 OPC 服务器使用，在编程时变量或符号的选择应能够保持一定的普适性。 其意思是，应用程序应可在多种场合下重复使用。

3.1.5 服务器和客户端 - 它们如何协作？

服务器和客户端配合工作

服务器和客户端在 COM 或 DCOM 的基础上进行通信。客户端并不是直接访问服务器，而是在 COM 库的协助下进行访问。通过指定 ProgID，OPC 客户端可直接寻址各个所需的 OPC 服务器。

客户端程序察觉不到通过 COM 访问与通过 DCOM 访问之间的差异。

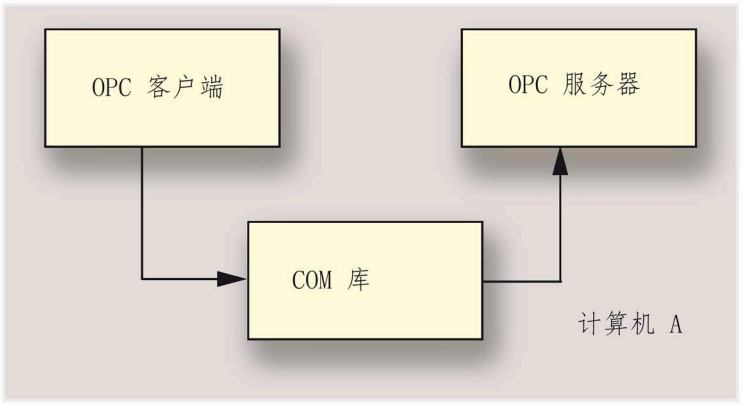

图 3-3 本地计算机上的 COM

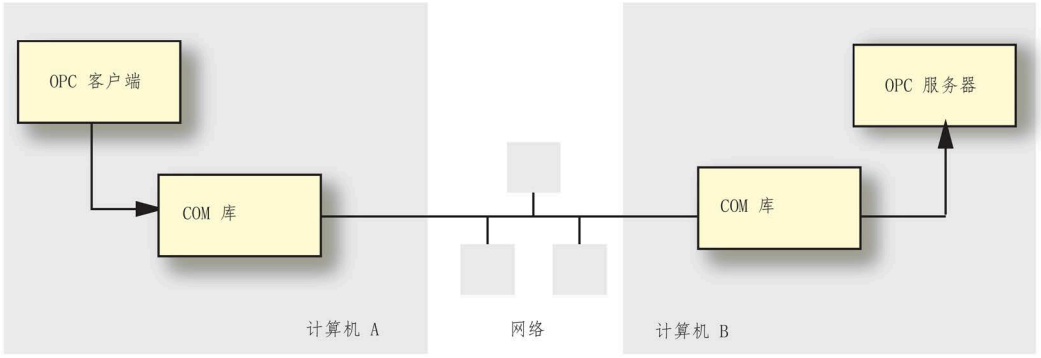

图 3-4 远程计算机上的 COM

应用了哪些属性和方法？

OPC 服务器的性能范围取决于其接口。因此，OPC 客户端了解待访问服务器的性能，从而能够选择可用的特定服务。从面向对象的角度而言，属性和方法代表了 OPC 服务器的服务。所有 OPC 服务器都有一组相同的属性和方法。除此之外，在 OPC 规范中，一些接口被标识为可选接口。如果服务器未提供这些可选的功能，客户端可识别这一情况并相应地作出反应。因此，不同制造商的组件可无故障地配合工作。

客户端可通过 OPC 接口在服务器上生成、使用和删除对象。OPC 客户端访问服务器功能并使用服务器的方法进行诸如读、写数据的操作。每个服务器功能都对应于客户端的一个调用。

3.1.6 基本术语

3.1.6.1 COM 对象 - 它们是什么？

为使客户端与服务器之间的协作更加有效，可以合并或指定类型相似的任务。 COM 对象将此实现。

什么是 COM 对象？

COM 对象是在 Windows 下运行的组件，它们通过其接口为其它组件提供定义的功能。一个 COM 对象可由多个应用程序同时使用。

什么是 COM？

COM 是 Windows 操作系统的中央组件，控制多个软件组件之间的交互。

使用 COM，OPC 服务器类似于 Windows 操作系统的一部分，因此与文件名、存储位置和版本无关。

OPC 机制的基础就是 COM，即源自 Microsoft 的“组件对象模型”。

COM 定义一项标准，该标准允许将对象定义为 Windows 中的独立单元以及超越过程界限访问这些单元。

可将 COM 对象理解为操作系统的扩展。 它们与编程语言无关，原则上可用于所有应用程序。

COM 对象的用户不能直接访问对象的数据和代码。

什么是 DCOM？

DCOM 是指"分布式组件对象模型"。 作为 COM 的深入开发，DCOM 支持分布式应用程序，允许网络中不同计算机上的软件组件之间进行协作。

COM 对象的结构

下图说明了具有四个接口的 COM 对象的结构。 只能通过接口访问对象。 访问由各种方法控制。 无法访问实际对象的整体或其所含数据或代码。

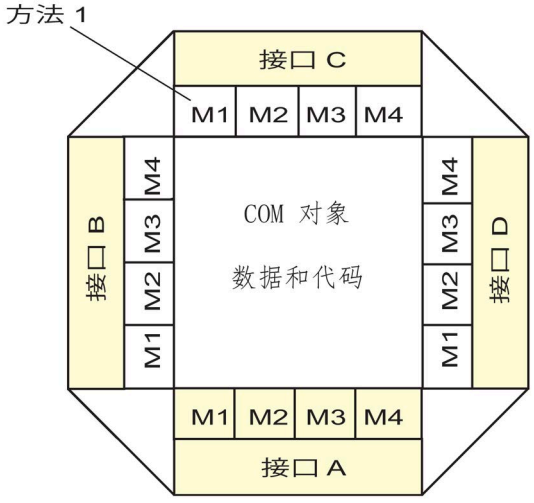

图 3-5　　COM 对象的结构

3.1.6.2 COM 对象，如何表示？

COM 对象的表示

文档中通常以图形表示 COM 对象。 对象特定接口显示在对象一侧，所有对象都附带的 IUnknown 接口显示在对象上边。

接口底层方法被接口掩盖。

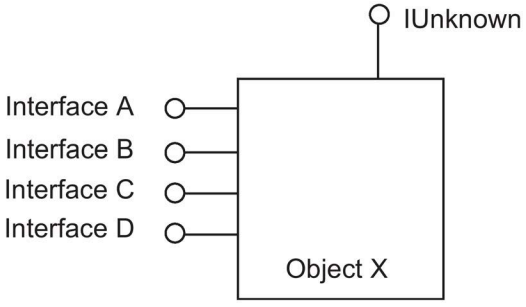

图 3-6 COM 对象的表示

3.1.6.3 COM 接口 - 有何作用？

COM 接口提供的功能

COM 接口是经过定义的、通常相关的一套方法，用于调用 COM 对象的功能。 它由引用方法的指针表组成。 COM 接口封装 COM 对象的功能，确保只能以定义的方式访问对象。 COM 接口都有唯一 ID，以使要访问 COM 对象的应用程序在访问前可以检查对象是否支持接口。

接口的结构

下图显示接口的基本结构。

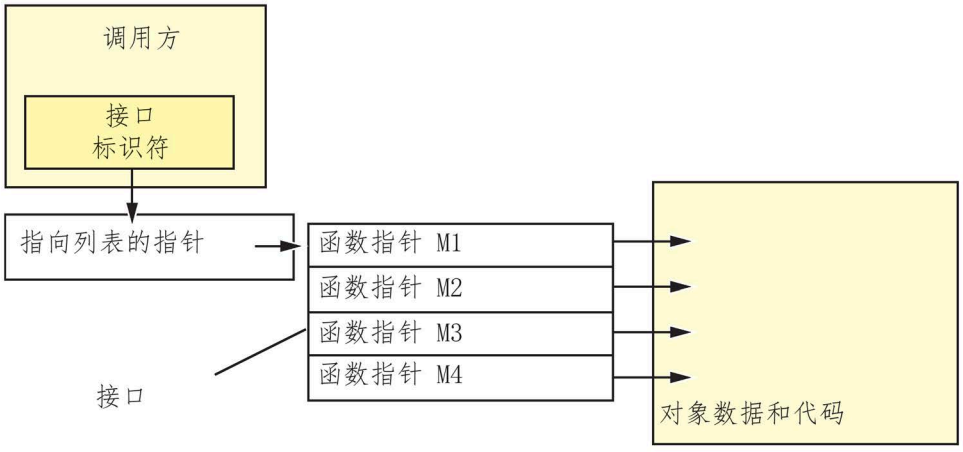

图 3-7 接口的结构

3.1.6.4 COM 接口类型 - 存在何种类型，如何访问？

存在以下这些接口类型

COM 区分两种接口类型：

- 自动化接口
- 自定义接口

这两种接口的区别在于内部方法调用。每种接口都有单独的接口规范。不过，它们同样适用于多种应用，如访问变量和接收信息。

自动化接口支持基于脚本语言（如 VB 或 VBA）的客户端应用程序。

自定义接口提高了基于 C 或 C++ 的应用程序性能。

自定义接口不适合基于脚本语言的开发工具的功能范围。通过自动化接口扩展 COM 对象，对象方法也可用于简单脚本语言。自动化接口使对象理解的调用可见于外部。

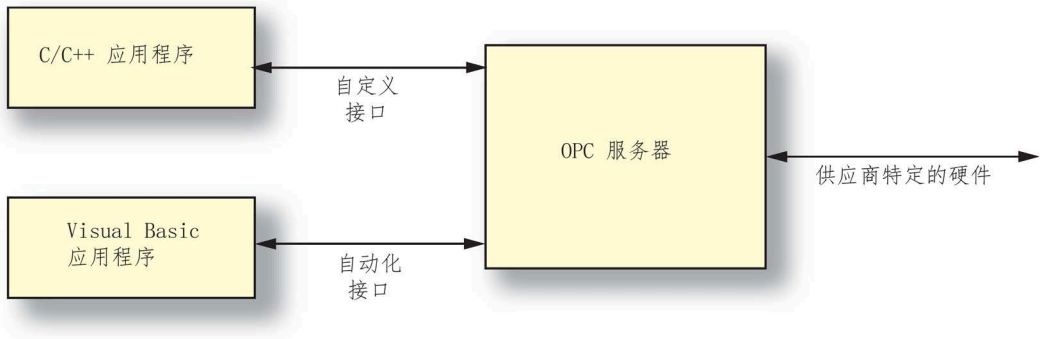

图 3-8 接口与应用程序之间的分配示例

.NET 客户端如何访问 COM 接口？

以下部分介绍使用自定义接口和自动化接口时的事件过程。

使用 OPC 自定义接口时的顺序

.NET 客户端可从托管代码内部访问自定义接口的常规对象。由于 COM 与 .NET 编程模型的属性不同（例如在 .NET 中没有指针访问），因此无法直接调用。

对于从托管代码到非托管代码的转换，必须使用 RCW（运行时可调用包装）。RCW 在托管 .NET 对象与非托管 COM 对象之间进行调解。

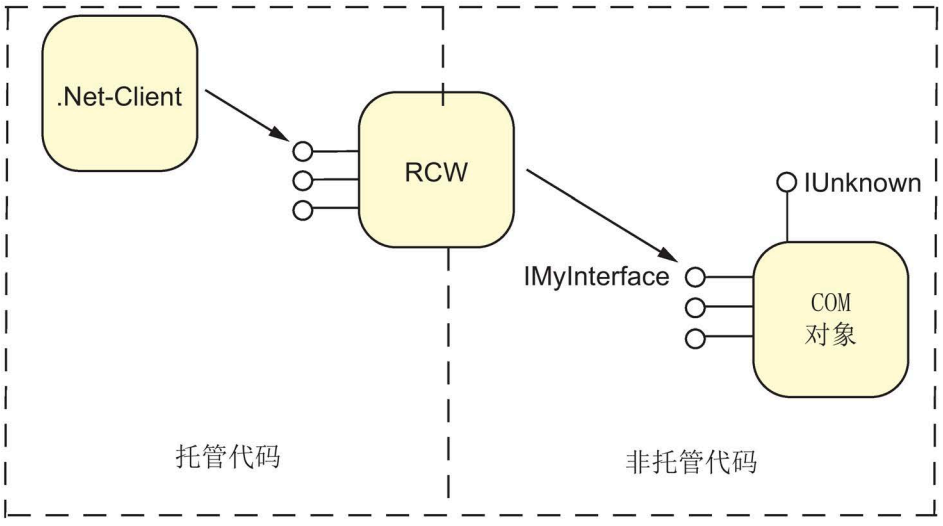

图 3-9　　使用自定义接口时的顺序

使用 OPC 自动化接口时的顺序

借助 .NET Framework 导入应用程序，创建所谓的互操作程序集（.NET 组件）。.NET 客户端可使用它来创建 COM 对象实例并调用 COM 对象的方法（如同 .NET 实例一样）。非托管代码（自动化接口）以此转换为 .NET 组件。

3.2 数据访问

3.2.1 数据访问接口简介

3.2.1.1 OPC 数据访问有何用途？

数据访问接口是与供应商无关的世界标准，用于读取、写入和监视过程数据。通信基于 Microsoft COM 协议。此标准已赢得用户和供应商的共同认可。用户程序范围从简单的办公应用程序延伸到复杂的 HMI（人机界面）或 SCADA（监控和数据采集）系统。

OPC 数据访问的用途

OPC 数据访问规范定义客户端与服务器程序间用于过程数据通信的接口。数据访问服务器允许一个或多个数据访问客户端透明访问多种数据源（如温度传感器）和数据宿（如控制器）。此类数据源和数据宿可位于直接插入 PC 的 I/O 卡上，也可位于控制器、通过串行连接或现场总线连接的输入/输出模块或类似设备上。数据访问客户端当然也可以同时访问多台数据访问服务器。

什么是数据访问客户端？

数据访问客户端可以是非常简单的 Excel 工作表或大型程序（如 Visual Basic）。然而，数据访问客户端也可以是大型程序的一部分。

什么是数据访问服务器？

数据访问服务器可以是简单程序，例如，通过串行接口访问 PLC 寄存器的程序。也可以是更复杂的程序，此类程序可以通过广泛的通信机制访问多台设备上的众多变量。数据访问服务器还可以是更大程序的一部分，使数据可用于这些程序。

3.2.1.2 OPC 数据访问 - 它是什么？

通过 OPC 数据访问 您可访问过程变量

数据访问是使用变量访问过程数据的 OPC 规范。 针对数据访问的 OPC 服务器管理过程变量和用于访问这些变量的的各种选项。 这使其能够：

- 读取一个或多个过程变量的值
- 通过写入新值修改一个或多个过程变量的值
- 监视一个或多个过程变量的值
- 指示值变化。

过程变量是必须在运行时采集的值的占位符。

3.2.1.3 OPC 数据访问的类模型 - 有何作用？

OPC 数据访问类模型的作用

数据访问的层级类模型在客户端访问数据时有助于调整所需时间和结果以适应当前应用程序的要求。数据访问分为三类：

- OPC 服务器
- OPC 组
- OPC 数据项

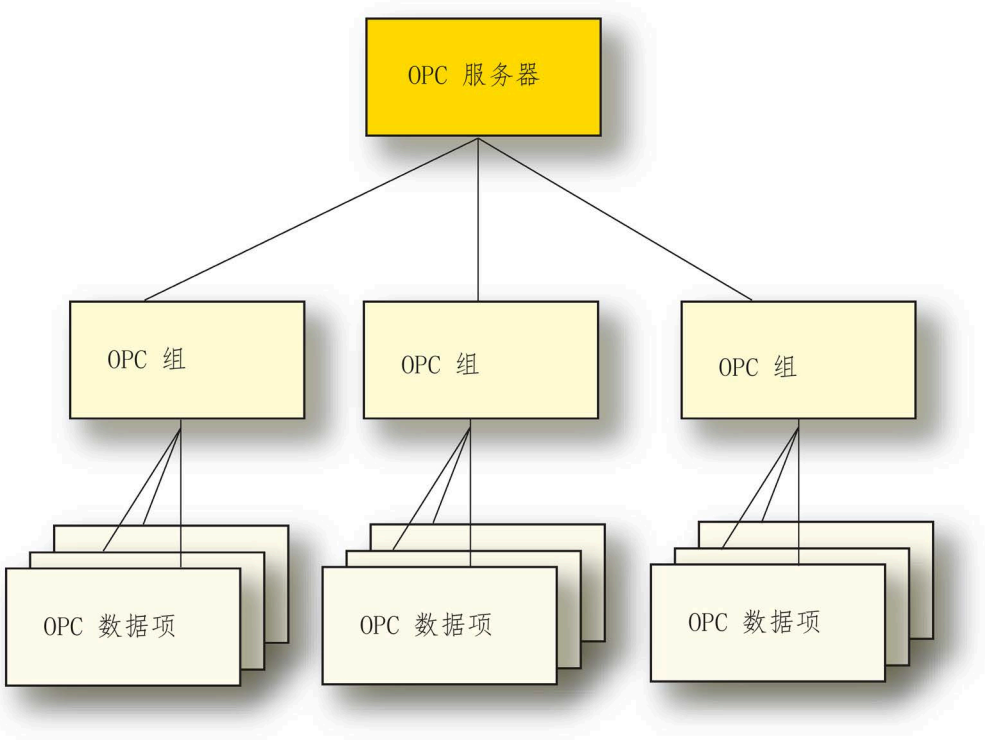

图 3-10 数据访问接口的类模型

客户端应用程序只使用操作系统的 COM 调用来生成 OPC 服务器类的对象。其它对象通过 OPC 服务器类或更低级别类的相关 OPC 方法进行创建。

类模型的适用范围是什么？

类模型既适用于自动化接口又适用于自定义接口。

3.2.1.4 OPC 服务器类 - 有何作用？

OPC 服务器类的作用

OPC 服务器类位于顶层。 每个 OPC 服务器都属于此类。 此类为数据访问服务器的所有其它服务提供访问。

借助于类特定属性和方法，您可获取可用过程变量的状态、版本和名称空间（可选）信息。 OPC 服务器类对象管理底层 OPC 组类的实例。

3.2.1.5 OPC 组类 - 有何作用？

OPC 组类的作用

OPC 组类是 OPC 服务器下层的类，用于构造 OPC 服务器使用的过程变量。OPC 客户端可以同时使用此类的多个对象。客户端可使用 OPC 组的对象来生成过程变量的适当单元并使用它们进行操作。例如，操作员监控系统中给定屏幕页面上的所有过程变量均可合并在同一组中。

OPC 组类定义可用于读写过程变量值的方法。

借助某些方法（读取和写入 OPC 数据项），可将多个变量合并到一个作业中同时传送。很多情况下，OPC 服务器还可以运行附加的内部优化。特别是在通过网络使用 OPC 服务器时，这些组操作允许高速执行。以较短的时间间隔连续执行单项作业通常造成性能下降。

自数据访问规范 3.00 起，可使用 OPC 组类设置 OPC 服务器的循环保持连接监视 (KeepAliveTime)。即使过程变量不变，OPC 服务器也会对 OPC 客户端调用核查函数（无数据值）。

3.2.1.6 OPC 数据项类 - 有何作用？

OPC 数据项类的作用

此类中的对象表示实际的过程变量，允许对单个数据项进行特定查询。每一个变量都是 OPC 服务器名称空间的一个元素（数据项），并由数据项 ID 标识。数据项 ID 由服务器制造商指定，在服务器名称空间中必须惟一的。每个数据项关联以下属性：

- 值
  上次获取的变量值
- 质量
  值的精度。如果质量好，确定获取该值。
- 时间标记
  获取变量当前值的时间。时间标记随着报给客户端的每次值变化而更新。如果变量的值不变，时间标记也保持不变。

变量起到什么作用？

要获取过程值，必须随 OPC 接口调用指定变量。通过指定变量，客户端可向服务器请求所需值。客户端需要通知服务器所需的每个变量，以决定将读取哪些变量。可以同步和异步读写变量。

客户端可将变量监视转交给服务器。变量值发生更改时，服务器向客户端发送一条相应消息。

服务器提供的变量可分为以下几种类型：

- 过程变量
  表示输入/输出设备的测量和控制量
  或
- 控制变量
  使用这些变量触发附加服务，例如密码传送。
  或
- 信息变量
  这些变量由通信系统和 OPC 服务器提供，而这些变量则提供连接状态和设备等信息。

以下为几个 OPC 数据访问服务器变量的示例：

- 可编程逻辑控制器的控制值
- 测量数据采集系统的数据
- 通信系统的状态变量

3.2.1.7 OPC 数据访问 - 有哪些接口规范？

OPC 数据访问有两种接口规范

数据访问的自动化和自定义接口已指定：

- 数据访问自动化接口，标准，1999 年 2 月 4 日，版本 2.02（及后续版本）
- 数据访问自定义接口，标准，2003 年 3 月 4 日，版本 3.00

有关规范的概述，请参见第 2 卷中的参考资料。

3.3 OPC 报警和事件

3.3.1 OPC 报警和事件简介

3.3.1.1 OPC 报警和事件 - 有何含义？

报警和事件

报警和事件是用于传送过程报警和事件的规范。它设计灵活，因此可用于多种事件源。适用范围从简单事件到复杂事件，甚至需要确认的事件。

OPC 规范定义状态图中条件事件的可能状态变化。

报警和事件有何用途？

举例来说，报警和事件服务器用于

- 识别事件 - 例如罐填料结束
- 判定事件状态 – 罐已满
- 确认事件 - 确认罐填料结束
- 监视确认 - 确认由罐报警信号设备监视，检测到报警后，报警信号即可关闭。

新事件也可通过信号进行指示而不确认。

标准化 OPC 报警和事件接口允许处理这些要求。

3.3.1.2 事件和事件消息 - 它们是什么？

事件

事件 是过程中必须向接收方发出信号的特殊状态。 OPC 客户端使用过滤条件设置哪些事件要向 OPC 客户端发送信号。

符合所选过滤条件的所有事件都必须从事件的产生者传送到用户。 这将报警和事件与数据访问区别开来。 变量监视期间，只对特定时间基准内的值更改发出信号。

事件消息

消息包含 OPC 规范中定义的参数，还可能包含供应商指定的关联值。

存在简单事件消息 和更复杂的状态相关消息。 对于这些复杂的状态相关消息，事件发送方可以要求 OPC 客户端确认。

事件类型

OPC 规范定义三种事件类型：

- 条件相关事件
  指示在 OPC 状态模型中定义的状态更改，与定义的条件有关。
- 跟踪事件
  指示过程更改，例如用户更改控制器设定值。
- 简单事件
  指示所有其它不涉及状态的事件，例如系统组件故障。

OPC 规范定义了用于接收消息的接口语法。 服务器提供何种事件类型由 OPC 服务器供应商指定。

3.3.1.3 OPC 报警和事件的类模型 - 有何作用？

OPC 报警和事件类模型的作用如下：

报警和事件的类模型允许调整 OPC 客户端以适应自动化解决方案的要求。报警和事件分成三类：

- OPC 事件服务器
- OPC 事件订阅
- OPC 事件区域浏览器

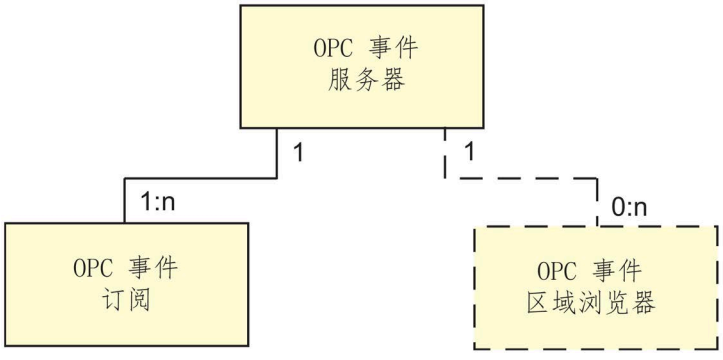

图 3-11　　报警和事件接口的类模型

3.3.1.4 OPC 事件服务器类 - 有何作用？

OPC 事件服务器类的作用

客户端使用 OPC 事件服务器类对象创建一个或多个 OPC 事件订阅类对象。 此类对象是事件组的订阅。 此类对象管理特定客户端需要的过滤器和属性。 通过过滤，客户端可指定要接收的事件。 SelectReturnedAttributes 方法可指定随每条事件消息返回的事件属性。 借助于 OPC 事件订阅类对象，客户端可创建实际组并执行组操作。

确认事件

使用 OPCEventServer 类的 AckCondition 方法，如果已在事件的 AckRequired 参数中指定，则客户端确认条件相关事件。 确认到达后，这使条件相关事件的 NewState 参数发生更改，因此产生新事件。

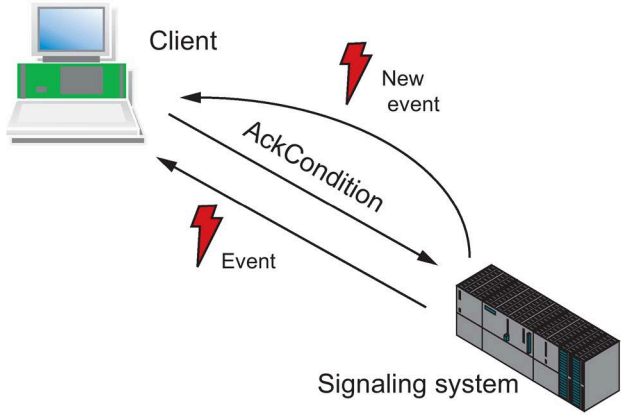

图 3-12　事件顺序和条件相关事件确认

3.3.1.5 OPC 事件订阅类 - 有何作用？

OPC 事件订阅类的作用

客户端使用 OPC 事件服务器类对象创建一个或多个 OPC 事件订阅类对象。 此类对象是事件组的订阅。 此类对象管理特定客户端需要的过滤器和属性。 通过过滤，客户端可指定要接收的事件。 可以指定随事件消息传送的事件属性。 借助于 OPC 事件订阅类对象，客户端可创建实际组并执行组操作。

过滤事件有哪些选项？

通过过滤，客户端可指定要接收的事件。 过滤器只是一个基于属性的事件定义。 这基于以下条件：

- 事件类型
- 类别
- 优先级
- 事件源

只有在与所有条件下的过滤器值匹配时，事件才转发到客户端。

为什么缓冲事件？

如果每个事件都单独传送到客户端，就需要比几个事件一起传送时更多的资源。 通过 BufferTime 参数，客户端可以指定事件只有在经过一定时间后才应该发送。 同时发生的事件在下一传送时间之前进行缓冲。

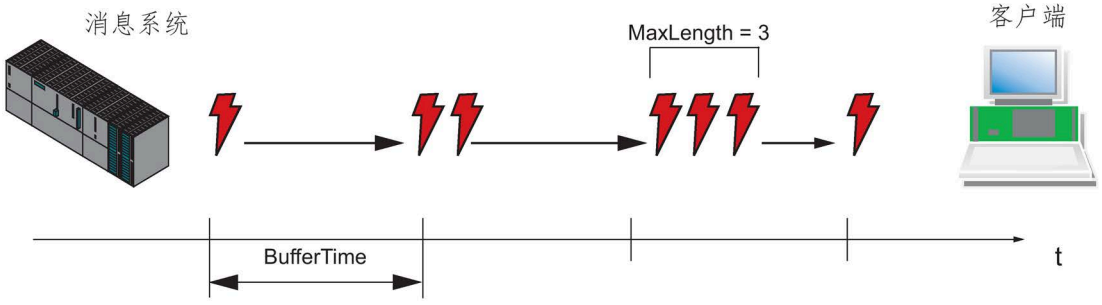

图 3-13 参数 BufferTime 和 MaxSize 的含义

可使用 MaxSize 参数指定要缓冲的最大事件数目。 达到指定数目后，所有事件都发送到客户端，而无论所选 BufferTime 间隔为何。

在 OPCEventServer 类的 CreateEventSubscription 方法和 OPCEventSubscription 类的 GetState 以及 SetState 方法中，BufferTime 和 MaxSize 用作参数。

3.3.1.6 OPC 事件区域浏览器类 - 有何作用？

OPC 事件区域浏览器类的作用

通过 OPC 报警和事件，您可将大型工厂分成多个区域。区域可用于过滤事件。借助于 OPC 事件区域浏览器类对象，您可对这些区域进行检查。

说明

OPC 事件区域浏览器类对象为可选，不受 SIMATIC NET 的 OPC 报警和事件服务器支持。

3.3.1.7 消息接收 - 工作原理是什么？

消息接收的工作原理

应用程序进行注册以接收消息（分为四步）：

顺序

- 客户端在服务器注册以接收消息。
- 客户端创建一个或多个 OPCEventSubscription 类对象。

- 客户端通过 IConnectionPointContainer 接口设置回调。
- 事件发生时，客户端提供一种由服务器调用的 OnEvent 方法。

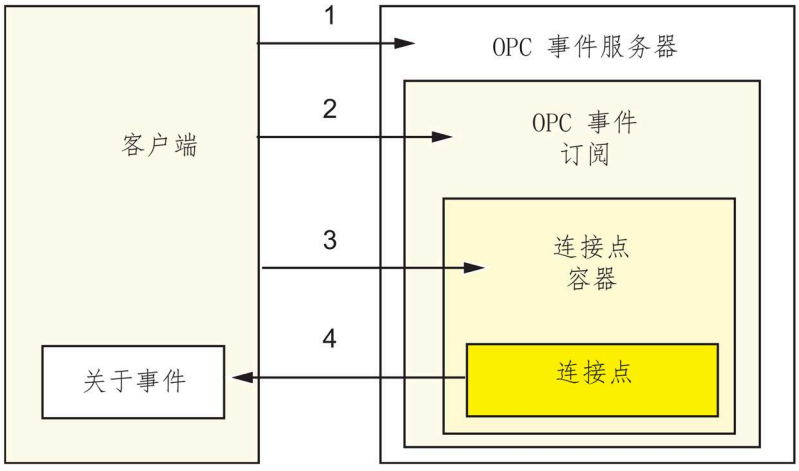

图 3-14　　用于接收消息的服务器和客户端连接

3.3.1.8　　SIMATIC S7 中的报警 - 如何定义？

如何定义报警

报警具有以下属性：

- 报警由二进制信号变化（沿）表示。
- 信号变化产生一种持续时间 t>0 的新二进制信号状态。
- 每次信号变化均可由信号接收方确认。
- 确认状态可由报警发起方监视。
- 在上次信号变化未经确认的情况下，信号可以再次变化。

SIMATIC S7 可通过各种块触发报警

表格 3- 1　　通过各种块触发报警

块	名称	监视的信号数	确认	关联值	严重性
SFB36	NOTIFY	1	否	1 ... 10	0 ... 127
SFB31	NOTIFY_8P	8	否	1 ... 10	0 ... 127
SFB33	ALARM	1	是 (SFB33)	1 ... 10	0 ... 127
SFB34	ALARM_8	8	是 (SFB34)	无	0 ... 127

块	名称	监视的信号数	确认	关联值	严重性
SFB35	ALARM_8P	8	是 (SFB35)	1 ... 10	0 ... 127
SFC17	ALARM_SQ	1	是 (SFC19)	1	无
SFC18	ALARM_S	1	隐式确认	1	无
SFC107	ALARM_DQ	1	是 (SFC19)	1	无
SFC108	ALARM_D	1	隐式确认	1	无

S7 用户程序指定报警接收方是否需要确认。S7 程序区分报警状态开始（报警进入的状态）确认和报警状态结束（报警退出的状态）确认。
OPC 接口不支持这种区分；仅支持报警发生确认。报警状态结束由报警和事件服务器隐式确认。

除块相关报警外，OPC 报警和事件服务器还支持：

- 符号相关报警 (SCAN)
  允许监视与 PLC 用户程序异步的 CPU 中 I、Q、M 和 DB 区域内的位。
- 诊断报警
  使用 WR_USMSG (SFC52) 的系统诊断和用户诊断

3.3.1.9 报警 - 会出现什么实际情况（示例）？

处理报警的示例

下面提供了处理报警的两个示例：

- 无确认报警
- 有确认报警

无确认报警

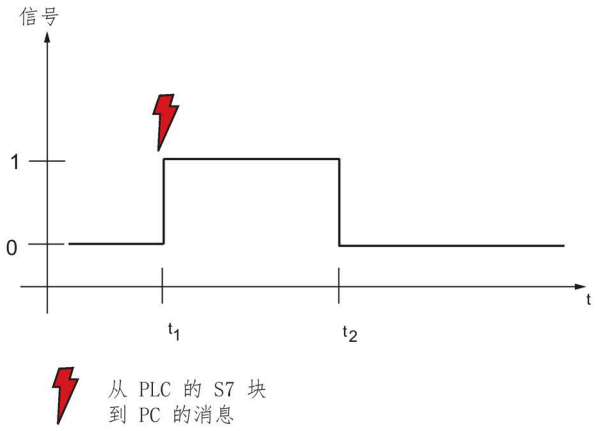

图 3-15 无确认报警的信号状态

S7 块监视生产期间填充的容器的液位。 容器填满时，S7 块触发报警 (t_1)，生产停止。 该报警无需确认，控制器无需采取任何其它措施生产便会中断。 控制器识别出容器已空时，它终止报警 (t_2)，生产恢复。

有确认报警

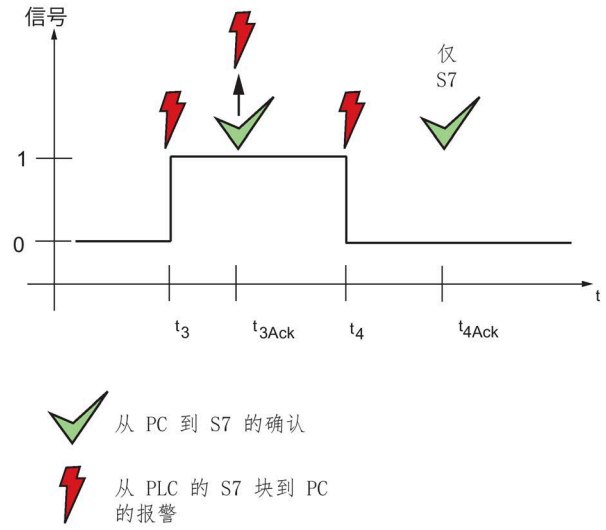

图 3-16　　有确认报警的信号状态

S7 块监视罐压力。 如果超过限值，S7 块触发报警 (t_3)，同时激活过压阀上的警示灯和报警器。

操作员的确认 (t_{3Ack}) 会关闭报警，但仍保持报警状态，因为罐压力高于限值。 确认并不关闭警示灯。 收到操作员的确认会在 S7 控制器上触发另一报警。

压力减小后，S7 块识别出当前压力低于限制值并终止报警 (t_4)。 报警状态结束也会触发一个报警。

操作员的报警状态结束确认会关闭警示灯 (t_{4Ack})。 在 OPC 接口上此确认不可见，因为 OPC 仅支持报警发生确认。

3.3.2 报警和事件接口

3.3.2.1 接口 - 为报警和事件指定哪些接口？

为报警和事件指定两个接口

为报警和事件指定自动化接口和自定义接口：

- 报警和事件自动化接口，标准，1999 年 12 月 15 日，版本 1.01
  OPC 报警和事件服务器的说明以及此服务器的自定义接口规范
- OPC 报警和事件自定义接口，2002 年 10 月 2 日，版本 1.10
  OPC 报警和事件服务器的自动化接口规范

有关规范的概述，请参见第 2 卷中的参考文献列表。

3.4 OPC 统一架构

3.4.1 OPC UA 简介

3.4.1.1 简介

是什么将 OPC UA 统一起来？

现有 OPC 标准之前的功能和选项（如数据访问、报警和事件、安全性、历史、复杂以及 XML 数据访问）均已结合到一套安全强大的新规范中：OPC 统一架构 (OPC UA)

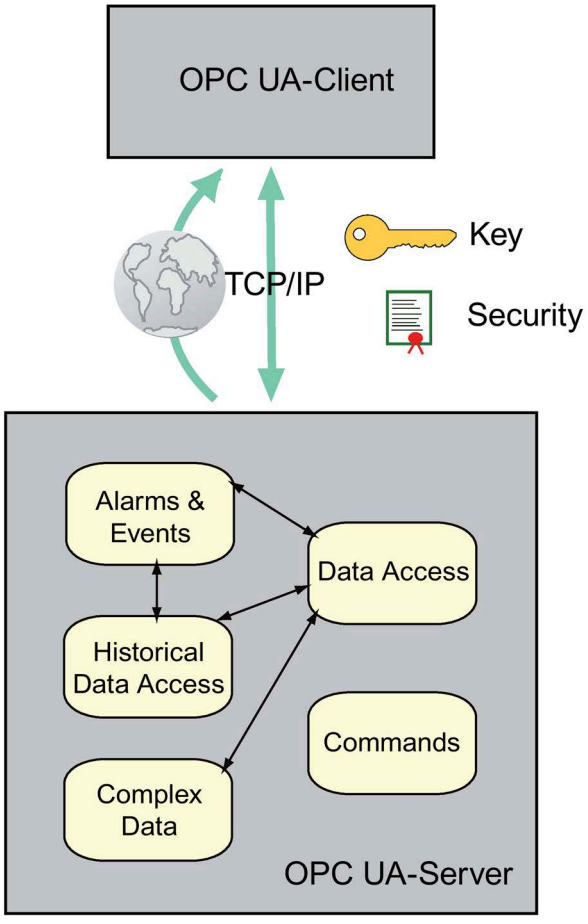

图 3-17 OPC 统一架构的功能

OPC UA 使用一套基于 TCP 的、安全强大且标准化的新通信协议。

OPC UA 架构有何优势？

- 将先前各项 OPC 标准统一成一个接口。

  这简化了客户端应用程序的开发。

- OPC UA 命名空间提供任意复杂程度系统的完整建模选项。与先前名称空间相比，其功能范围显著扩大。

与 COM/DCOM 接口相比，OPC UA 通信有何优势？

- 与平台无关的通信
- 因独立于 DOM 而更安全：
  - 没有复杂的安全设置
  - 没有开放端口 135
  - 不再有动态分配的端口
  - 防火墙设置更加简易
  - 无长时间超时和干扰
- 使用基于证书的现代验证程序，提高了数据安全性
- 针对特定应用程序，选择不同的适用通信协议

3.4.1.2 OPC UA 的安全性

如何确保安全？

OPC UA 客户端和 OPC UA 服务器必须使用数字证书和相应密钥互相验证和授权。消息进行相应加密。使用 X.509 证书进行验证。

通常为 OPC UA 客户端应用程序分配证书存储器。经过授权的 OPC UA 服务器的密钥可以储存于此。

SIMATIC NET OPC UA 服务器使用客户端上的目标系统特定 Public Key Infrastructure (PKI) 管理证书。

如何授权证书管理？

OPC UA 定义了可包含证书管理器的 Global Discovery Service (GDS)，此管理器可接管 OPC UA 应用程序（服务器和客户端）的证书、Trust List 和 Certificate Revocation Lists, CRL 的管理和分发。

3.4.1.3 OPC UA 的通信类型

OPC UA 通信类型"TCP 二进制"和"XML"涉及什么内容？

在最低层，OPC UA 的通信协议基于 TCP，因此在嵌入式系统中也可以跨平台使用。任何情况下都要求安全加密传输。

根据标准，OPC UA 接口提供以下协议选项：

- 通过端口 80/443 使用 HTTP/HTTPS 的简单 XML/SOAP
- 通过端口 4840（及其它端口，如端口 55101，如果添加其它服务器，也可能是端口 55105）的二进制 TCP

更好的是：封包二进制 TCP

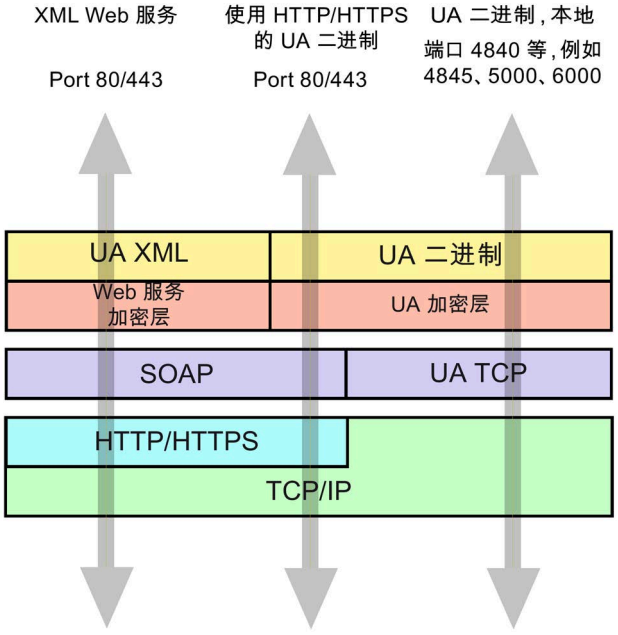

图 3-18 OPC UA 使用的协议

可以使用 OPC UA 用户界面上的 OPC UA 服务器的 URL 地址来选择协议。您有以下两种选项。

示例：

- OPC UA XML Web 服务，例如通过指定 URL：
  - http://<hostname>:80

    或
  - https://<hostname>:443
- 纯（原生）二进制 TCP 协议，通过指定：
  - opc.tcp://<hostname>:4840

在应用层，OPC UA 功能调用完全相同。

并非所有 OPC UA 服务器都支持所有协议。

OPC UA 原生二进制协议有何优势？

在 OPC UA 中，“OPC UA 原生二进制”协议的传输速度最高，因为数据经过压缩后再传输，而且几乎不需要使用封包信息。它需要最少的额外工作。例如，不像 SOAP 和 HTTP 那样需要 XML 解析程序。

格式标准化程度深至二进制层。这稳定了 OPC UA 客户端与服务器的数据交换，因为 XML 中不允许存在空格或注释等。

对于“OPC UA 原生二进制”协议，使用专门指定的 TCP 端口 4840 进行通信，而对于 SIMATIC NET OPC 服务器，还可根据优先级使用端口 55101 到端口 55105。可在防火墙中启用或禁用这些端口。

XML Web 服务协议有何优势？

XML 非常容易和 OPC UA 应用程序的公共开发环境结合使用。

防火墙通常已经过设置，为 HTTP 启用端口 80，为 HTTPS 启用端口 443，也可在防火墙中启用这些端口。对于 XML Web 服务的使用来说，这意味着通常无需额外组态便可实现 Internet 访问。

OPC UA 应用程序可以使用哪些编程语言来对 OPC UA 接口进行寻址？

OPC UA 客户端可通过 C、.NET（C#、VB.NET）、JAVA 和 C++ 接口访问 OPC UA 接口。OPC 基金会提供相应的库和汇编，包括通信堆栈。

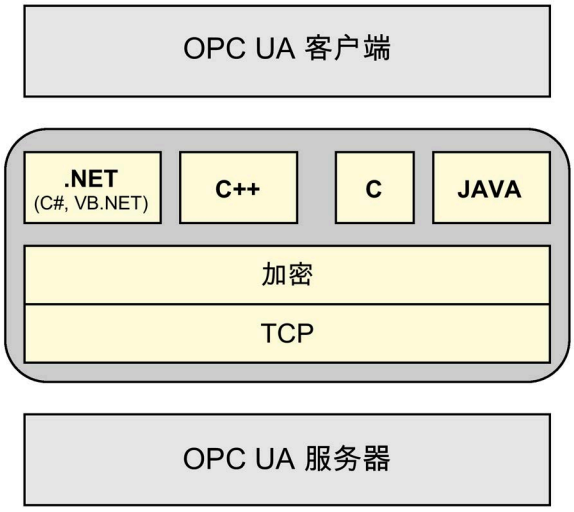

图 3-19　　OPC UA 客户端借助各种编程语言访问 OPC UA 服务器

3.4.1.4 OPC UA 的名称空间

OPC UA 命名空间包含哪些内容？

OPC UA 的名称空间不再只由文件夹、数据项和属性组成。它是一个带有附加信息和链接的节点网络。

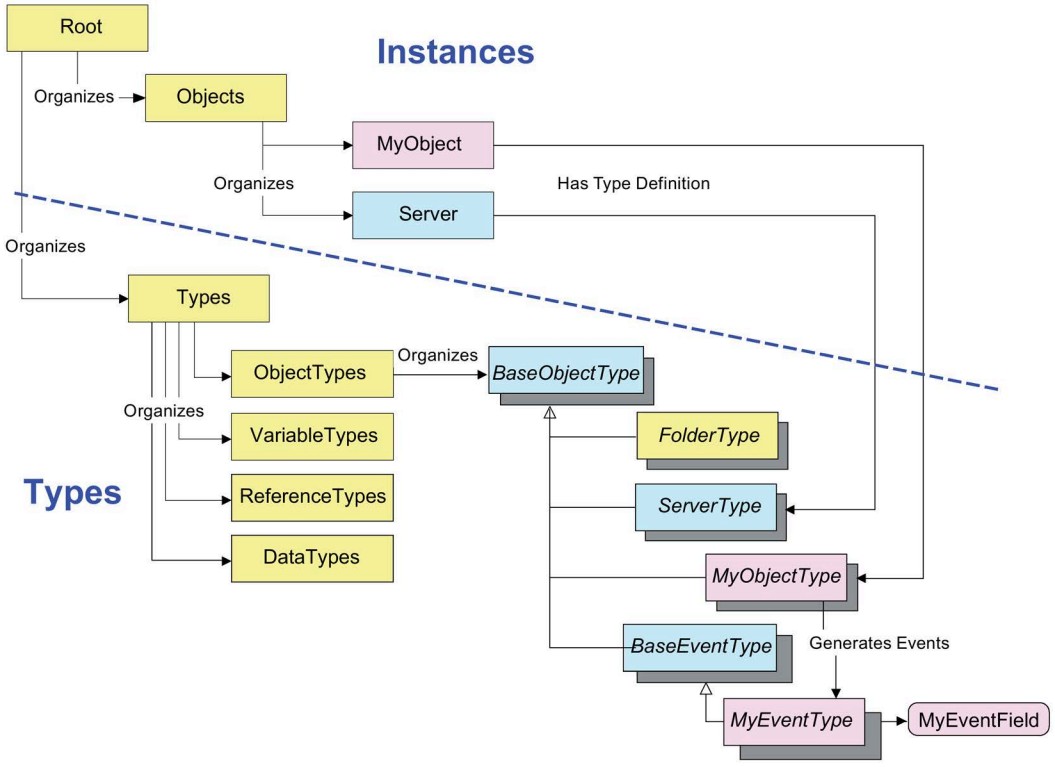

图 3-20 OPC UA 命名空间的结构

方框表示节点。这些是 OPC UA 的对象。箭头表示源节点到目标节点的引用。

节点既用于用户数据（实例），也用于其它信息，如数据的类型描述（类型）。OPC UA 的节点可分为：

- 类型

  这些是在 OPC UA 规范中指定的节点类型，也可能由相关供应商指定，它们按其属性唯一定义。共有以下四种基本类型：

  – ObjectTypes

  – VariableTypes

  – ReferenceTypes

  – DataTypes

  它们定义其它类型，其中一些显示在图中“ObjectTypes”类型右侧。

  这些类型用作实例的类型描述。

- 实例

  这些是实际项目的对象实例。根据节点类型，它们通过引用各种类型获取属性。在图中，“MyObject”和“Server”两个对象引用“MyObjectType”和“ServerType”两种类型。

OPC UA 服务器的根组织类型和实例。这种组织还包括其它节点的定义。

节点可有以下属性：

- 可读取的属性
- 可调用的方法
- 可报告的事件

OPC UA 规范中有很多标准节点。可以添加特定供应商的其它节点类型。命名空间可显示在 OPC 客户端中。

通过“检测”在系统中查找 OPC UA 服务器

OPC UA 搜索服务“发现”允许在系统中查找 OPC UA 服务器。发现服务使用为 OPC UA 保留的端口 4840、主机名称和/或 IP 地址。它报告所有 OPC UA 服务器的端点、协议、端口和安全要求。

OPC UA 服务器的端点

OPC UA 服务器提供通信端点。

端点是网络中的物理地址，允许 OPC UA 客户端访问 OPC UA 服务器的一项或多项服务。

有关 OPC UA 服务器端点的详细信息，请参见“SIMATIC NET PC 软件”DVD 中该手册的第 2 卷。

3.4.1.5 OPC UA 的其它特性

OPC UA 还有什么作用？

OPC UA 还提供多种其它功能，以下对其中一些进行简要介绍：

- 冗余

  OPC UA 提供多个 OPC UA 客户端和多个 OPC UA 服务器之间的冗余功能，从 OPC UA 客户端之间简单的会话转移直到 OPC UA 服务器之间的协调冗余。

- 连接监视

  OPC UA 服务器检测 OPC UA 客户端连接断开，OPC UA 客户端检测 OPC UA 服务器连接断开。标准已指定监视时间。

- 数据值存储

  连接断开一般不会造成数据丢失。数据值都已储存。可以再次请求未正确接收的数据。数据接收可以确认。

3.4.2 OPC UA 接口

3.4.2.1 OPC 统一架构有何接口规范？

OPC 统一架构有何接口规范？

OPC UA 规范由多个部分组成。这些是服务规范。

用户接口由和语言相关的 OPC UA 服务器堆栈或 OPC UA 用户库指定，其同样由 OPC 基金会提供。规范由以下部分构成：

部分	主题	标题
第 1 部分	概念	OPC UA 规范：第 1 部分 – 概念，版本 1.0 或更高
第 2 部分	安全模型	OPC UA 规范：第 2 部分 – 安全模型，版本 1.0 或更高
第 3 部分	地址空间模型	OPC UA 规范：第 3 部分 – 地址空间模型，版本 1.0 或更高
第 4 部分	服务	OPC UA 规范：第 4 部分 – 服务，版本 1.0 或更高
第 5 部分	信息模型	OPC UA 规范：第 5 部分 – 信息模型，版本 1.0 或更高
第 6 部分	服务映射	OPC UA 规范：第 6 部分 – 映射，版本 1.0 或更高
第 7 部分	配置文件	OPC UA 规范：第 7 部分 – 配置文件，版本 1.0 或更高
第 8 部分	数据访问	OPC UA 规范：第 8 部分 – 数据访问，版本 1.0 或更高
第 9 部分	报警和条件	OPC UA 规范：第 9 部分 – 报警和条件，版本 1.0 或更高
第 10 部分	程序	OPC UA 规范：第 10 部分 – 程序，版本 1.0 或更高
第 11 部分	历史访问	OPC UA 规范：第 11 部分 – 历史访问，版本 1.0 或更高
第 12 部分	发现	OPC UA 规范：第 12 部分 – 发现，版本 1.0 或更高
第 13 部分	执行器	OPC UA 规范：第 13 部分 – 执行器，版本 1.0 或更高
第 14 部分	PubSub	OPC UA 规范：第 14 部分 – PubSub，版本 1.0 或更高

有关该规范的详细概述，请访问：OPC 基金会 (https://opcfoundation.org/developer-tools/specifications-unified-architecture)

3.4.2.2 OPC UA 客户端服务器

如何建立与 OPC UA 服务器的连接？

术语定义

术语	含义
安全通道	用于在 OPC UA 客户端与服务器的通信堆栈之间进行安全数据传输的通信通道。每条安全通道都有一个全局标识符并包含通过此通道发送的消息加密具体信息。
通道标识符	建立通道时指定的安全通道标识符。
通信堆栈	在应用层和硬件层之间构建的一组软件模块，用于处理节点之间通信期间的各种任务。
证书	在此：标识加密系统内节点的密钥（签名）。
会话	限时会话，其间 OPC UA 客户端与服务器交换数据。
会话 ID	建立连接后 OPC UA 服务器分配给 OPC UA 客户端的会话标识号。对于所有后续查询，客户端必须将该会话 ID 通知服务器。

建立连接

下图显示了在 OPC UA 客户端与 OPC UA 服务器之间建立连接所涉及的组件概览。

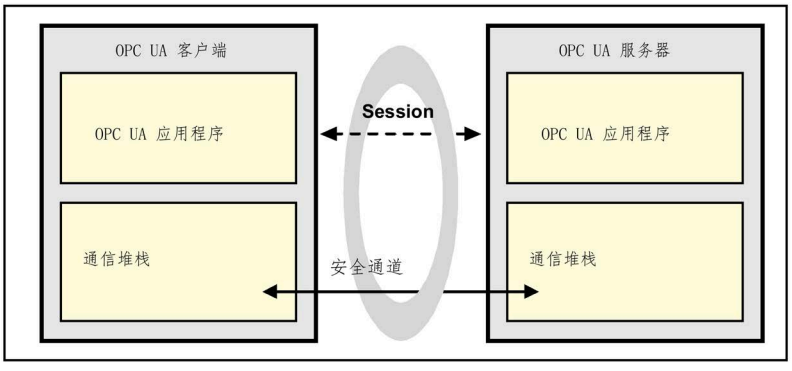

图 3-21 OPC UA 中的连接建立过程

通过以下步骤建立连接：

1. 建立安全通道

   使用服务“OpenSecureChannel”建立安全通道。

   此服务在会话期间打开或更新安全通道。安全通道允许在 OPC UA 客户端与服务器之间进行保密信息传输。

   建立安全通道后，通信堆栈对要发送的帧应用多种安全算法。在获得发送方证书签名后，帧被加密传输，只有授权的伙伴才能解密帧。

2. 建立会话

   使用“CreateSession”服务建立会话。

   OPC UA 客户端借助于此项服务建立会话。OPC UA 服务器返回会话 ID。

   对于后续的客户端查询，只有当客户端同时传输通道 ID 和会话 ID 时，服务器才会接受查询。

如何浏览 OPC UA 名称空间？

以下服务可用于浏览 OPC UA 名称空间：

- “浏览”

  此服务用于确定节点的引用（链接）。

- “读取”

  此服务用于确定一个或多个节点的一个或多个属性。

响应提供查询的值（引用、属性 (property) 或属性 (attribute)）。

如何读写数据？

如何实现简单读写？

“读取”和“写入”两项服务可用于读写节点的属性值。

- “读取”

  此服务用于获取一个或多个节点的一个或多个属性。 通过元素索引方式类似于数组的结构化属性值，客户端可读取全部索引值，它们可读取特定区域或单个元素。

  值的新旧程度由“maxAge”参数确定。

- “写入”

  此服务用于将值写入一个或多个节点的一个或多个属性。 通过元素索引方式类似于数组的结构化属性值，客户端可写入全部索引值，它们可写入特定区域或单个元素。

  在值已写入或识别出值无法写入之前，服务作业保持待定状态。

“读取”和“写入”访问使用节点的“NodeID”。 NodeID 是 OPC UA 名称空间内节点的标识符。

如何监视 OPC UA 数据和事件？

术语定义

术语	含义
订阅	订阅用于从 OPC UA 服务器到客户端传输数据。订阅包含一组以通知形式传送给客户端的监视数据项。
监视数据项	客户端定义监视数据项以获取数据和事件。监视数据项标识要监视的数据项、与其对应的订阅和通过订阅传输数据的通知。
数据项	数据项可以是任意节点属性。
通知	描述数据值或事件变化的数据结构。此数据结构以监视数据项的数据填充。
通知消息	订阅将通知封装在通知消息中传送给客户端。
发布请求	客户端发给服务器的数据传输请求
属性	OPC UA 规范定义的简单节点特性。
节点/ NodeID	节点是名称空间的基本组成部分。每个节点都由其 NodeID 标识。

MonitoredItem 模型

MonitoredItem 模型描述以下属性或对象的监视：

- 属性

  监视属性值是否更改。 属性的每次更改都会生成一条通知（不使用过滤器，见下文）。

  不要将属性与变量的值属性互相混淆。

- 变量

  变量可以更改值或状态。 与上面提到的"属性"不同，对于变量，监视变量的"值属性"（状态）。

- 节点

  节点可以提供值和事件。 事件只能由节点构成（"SubscribeToEvents"位已在"EventNotifier attribute"中置位）。 可使用对象和视图监视事件。

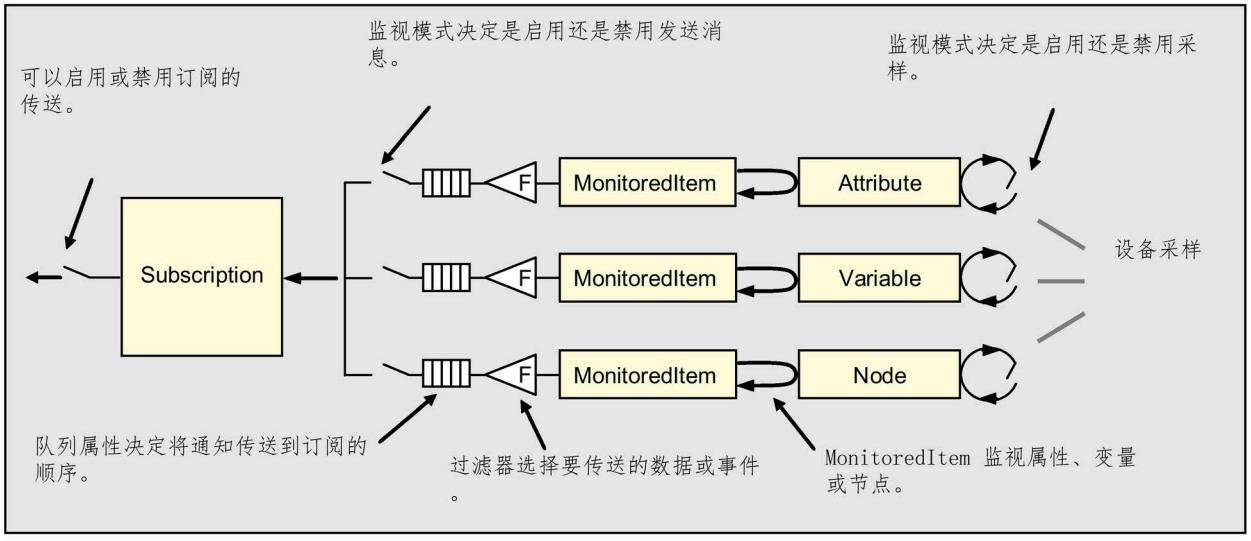

图 3-22　　MonitoredItem 模型

使用监视数据项模型监视数据概述

监视数据时，信息按以下步骤通过 OPC UA 服务器从设备传送到 OPC UA 客户端：

1. UA 客户端为每个要监视的数据项定义一个监视数据项。
2. OPC UA 服务器监视被扫描设备数据项的属性、变量或节点，最新数据存储在监视数据项中。
3. 每个监视数据项生成一条通知（如果监视数据项的监视模式启用了该功能，参见"监视数据项的属性"）。
4. 订阅将通知集合为一条通知消息。
5. 订阅将通知消息传送到客户端。
6. 客户端确认接收了通知消息。

MonitoredItem 的属性

MonitoredItem 有四种属性，各具以下功能：

- 采样间隔

  客户端生成的每个 MonitoredItem 都分配一个采样间隔。 采样间隔指定服务器对低级数据源（设备）采样的最短间隔。

  采样间隔继承自订阅的发布间隔（参见下文），或单独组态覆盖发布间隔。 默认情况下，采样间隔具有发布间隔的值。

- 监视模式

  监视模式指定通知的采样和传送是处于启用还是禁用状态。

- 过滤器

  过滤器定义传送的值或事件的变化幅度。 过滤器还可过滤事件的"EventType"属性，如 EventID、EventType、SourceNode、时间和描述。

  不对属性应用过滤器，因为属性的每次更改都会产生一条通知。

  过滤器还检测派生自节点的事件是否已传送到客户端。

- 队列属性

  队列属性可用于指定向客户端传送的顺序。

具有订阅的数据传输

订阅用于将通知传输至客户端。

订阅具有一个或多个由客户端分配的 MonitoredItem。 MonitoredItem 生成由订阅集成到 NotificationMessage 的通知。 订阅将一条或多条 NotificationMessage 传输到客户端。

使用“CreateSubscription”服务创建订阅。 这具有下列特性：

- 发布间隔

  订阅具有决定订阅变为激活状态的周期的发布间隔。 在此发布周期内，订阅尝试将 NotificationMessage 发送至客户端。

  NotificationMessage 包含尚未发送至客户端的通知。

- 对“发布请求”的回答

  将 NotificationMessage 作为对发布请求的响应发送至客户端。 在从服务器接收发布请求后，即会在会话的队列中输入此发布请求。

  – 在通知已准备就绪可以进行传输时，由属于当前会话的订阅将发布请求从队列中移除并进行处理。
  – 如果没有已准备就绪可以进行传输的通知，则发布请求将不会从会话的队列中移除，并且服务器会一直等到下一个周期，然后检查是否存在通知。

  在通知已存在但发布请求尚未存在的周期开始时，服务器将变为等待发布请求的状态。 在系统收到发布请求后，即会处理此请求而不必等待下一个发布间隔。

- NotificationMessage 的序列号（丢失的消息）

  每条 NotificationMessage 都具有消息丢失时允许客户端识别的单独序列号。

- 保持激活状态的计数器

  订阅具有保持激活状态的计数器，可对没有可供传输的通知的连续周期进行计数。 达到计数器的最大可选值时，发布请求将从队列中移除并用于发送保持激活状态的消息。 保持激活状态的消息将告知客户端服务器仍处于激活状态。

  保持激活状态的消息是对发布请求的响应，其中 NotificationMessage 不包含通知，而是包含下次将要发送的 NotificationMessage 的序列号。

- 启用“发布”服务

  创建订阅时，客户端可启用或禁用订阅的“发布”服务。 或者，也可使用“SetPublishingMode”服务启用/禁用“发布”。

  禁用“发布”后，订阅不会向客户端发送任何 NotificationMessage，不过，它会周期性地变为激活状态并将保持激活状态的消息发送至客户端。

- 有效期计数器

  订阅具有有效期计数器，而该计数器对没有客户端发布请求的连续发布周期进行计数。 当计数器达到根据“创建订阅”服务的“MaxKeepAliveCount”参数为订阅有效期所计算的值时，订阅将会关闭。

  如果关闭订阅，则会删除其 MonitoredItem。 服务器还会发送具有“代码 Bad_Timeout”状态的通知消息“StatusChangeNotification”。

- 确认 NotificationMessage 和通知缓冲区

  订阅具有用于 NotificationMessage 重复传输的缓冲区。 NotificationMessage 将保留在此缓冲区中，直到客户端对其进行确认，但至少持续 1 个保持激活状态的间隔。

“发布”服务

“发布”服务的用途共有两个：

- 请求服务器发送 NotificationMessage 或保持激活状态的消息
- 确认一个或多个订阅的 NotificationMessage 的接收

因为发布请求并非针对特定的订阅，所以任何订阅都可使用它们。

注册后，如何实现特别快速的读取和写入？

要快速地读取和写入已注册节点的属性值，可在注册相关节点后使用“Read(..,handle,..)”和“Write(..,handle,..)”方法。 利用这些方法，可通过短暂调用已注册节点来实现节省时间的数据传输。 访问将通过已注册节点的 NodeID 来进行。

读取/写入将按照下列步骤操作：

1. RegisterNodes()
2. Read(..,handle,..)

   Write(..,handle,..)
3. UnregisterNodes()

在 OPC 数据访问中，“RegisterNodes”方法的功能与“AddItems”方法的功能类似。

事件、条件和报警的工作原理是什么？

本部分介绍事件、条件和报警。

事件

在过程中，事件描述需要报告给接收方的特殊状态。OPC 客户端使用过滤条件来选择已报告给 OPC 客户端的事件。

OPC UA 事件与先前的 COM 接口 OPC 报警和事件上的事件不同，在后者中，用于 OPC UA 事件的访问技术和接口与用于 UA 数据访问的相同。

条件

条件源自常规事件。条件用于表示系统或系统组件之一的状态。下面显示了几个示例：

- 温度超过组态值。
- 设备需要维护。
- 在继续进行接下来的处理步骤之前，需要用户对批处理进行确认。

条件的主要状态为"启用"和"禁用"。"禁用"状态用于通过服务器关闭条件。"启用"状态通常由附加的子状态进行扩展。

切换到"禁用"状态将导致条件事件。但是，在条件重新切换为"启用"状态之前，不会生成更多的事件报警。

在条件切换为"启用"状态后，该切换以及所有后续切换将导致服务器生成条件事件。

条件对"启用"状态有效。OPC 服务器和设备会处理条件。

条件对"禁用"状态无效。OPC 服务器和设备无需处理条件，不存在此条件的"事件"报警。

确认

AcknowledgeableConditionType 类型源自 ConditionType 类型。
AcknowledgeableConditionType 类型包含条件的子状态以指示是否需要对条件进行确认。

报警

报警是条件的特殊形式。AlarmConditionType 是 AcknowledgeableConditionType 的特殊形式，其具有将被添加至条件的“启用”状态、复位状态和抑制状态的概念。

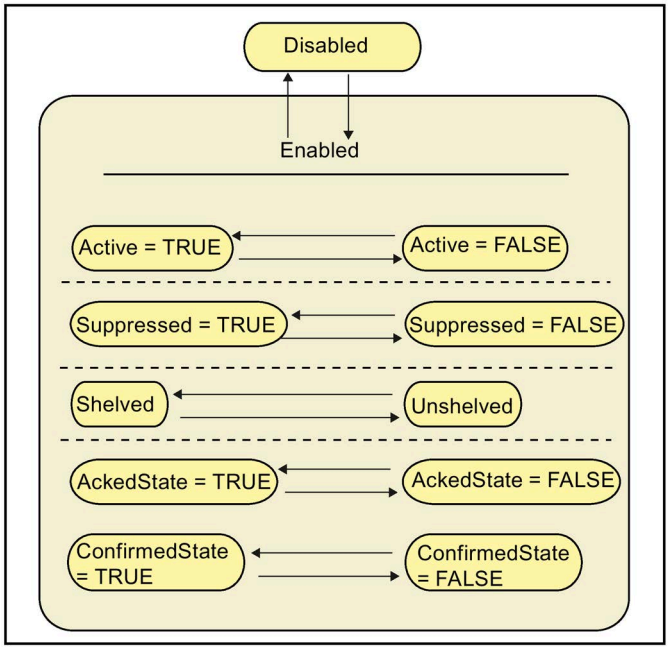

图 3-23　　报警的状态模型

处于“启用”状态的报警将指示由条件表示的状态当前已存在。如果报警处于“禁用”状态，则这将指示状态已返回到正常状态。

报警的一些子类型启动“启用”状态的子状态。示例：
报警表示温度具有高级状态和临界高状态。

OPC 报警的实际的源可以是来自 SIMATIC S7 的报警，请参见“SIMATIC S7 中的报警 - 如何定义？ (页 111)”部分

名称空间中的条件实例

因为条件始终具有“启用”或“禁用”状态，并且还可能有多个子状态，所以具有标识名称空间的条件实例切实可行。如果服务器代表条件实例，则这些实例将在名称空间中显示为“拥有”它们的对象的组件。例如，具有集成的高温监视功能的温度传感器会在名称空间中显示为具有对 LimitAlarmType 的 HasTypeDefinition 引用的多个温度传感器对象实例。温度传感器对象具有对条件的 HasCondition 引用。温度传感器对象还可具有对条件实例的 HasComponent 引用。

报警访问的客户端将建立具有报警功能的对象（例如通知温度传感器对象）的“EventNotifier”属性订阅，并会被告知状态变化。通过建立条件实例的相关属性的“值”特性订阅，实例的可用性允许客户端监视条件的当前状态。但是，在这种情况下，可能不会将所有状态变化告知该客户端。

即使不能始终提供位于名称空间中的条件实例，此机制也会允许对特定条件实例进行直接访问（读取、写入和方法调用）。例如，如果条件实例未显示，则将无法调用特殊条件实例的“启用”或“禁用”方法。

更多相关信息，请参见 OPC 报警和事件 (页 106)部分

如何接收事件、条件和报警？

UA 客户端可使用与数据监视技术相同的技术来接收事件。名称空间可进行浏览，具有 HasNotifier、HasEventSource 和 HasCondition 引用的节点指示了事件或条件可用。然后，UA 客户端便会注册这些节点的订阅。要接收事件，必须随后设置适合的过滤器。

3.5 SIMATIC NET 中的 OPC 数据访问及 OPC 报警和事件的性能

3.5.1 性能 - 我如何才能对其进行最佳利用？

对于 COM inproc 服务器，可提高性能

在某些情况下，例如使用基于 PC 的控制器时，必须对过程数据进行极其快速的访问。

在基于 COM 的客户端服务器架构中使用 OPC 时，的确会涉及某些内部执行时间，而这取决于 OPC 服务器的执行情况。

在出于过程切换的原因使用本地服务器（也称为"进程外服务器"；具有其自己进程空间的 EXE 文件），并从客户端向服务器传输函数参数（封送）时，这些时间将导致主要问题。

如果 OPC 服务器被作为进程内服务器实施，则可避免用于更改进程和封送的时间，因为 OPC 服务器会采用动态链接库 (DLL) 的形式，并在客户端的进程空间中运行。

然而，使用进程内服务器的确具有一些缺点，在选择服务器时必须予以考虑：

任何时候都只能有 1 个客户端使用服务器。如果多个客户端同时使用进程内 OPC 服务器，则将意味着会在不同的进程空间中多次生成服务器，并会导致对相同硬件的同时但不协调的访问。因此，只有首先发起的客户端访问才能访问过程数据，而其它客户端则会被拒绝访问。

OPC 服务器的稳定性取决于客户端。如果 OPC 客户端以不受控制的方式运作，并造成访问违例，则 OPC 服务器也将受到影响。结果是 OPC 服务器无法根据需要复位通信模块。此外，也会无法使用组态程序来显式关闭 OPC 服务器。

对于速度极快的 DP 协议，SIMATIC NET 提供了进程内服务器，而该服务器实际上也为 OPC 客户端提供了 DP 协议的全部性能。

甚至提高多个客户端的性能？

同样可以。如上所述，高性能进程内服务器只能供一个客户端使用。如果性能要求更高，为使两个或两个以上客户端能同时使用服务器，则应提供另外一个组态变量。要使用此变量，所有底层 DP、SR 或 S7 协议库以及作为进程内服务器的 COM 服务器均应在进程外 OPC 服务器上进行加载。如果在 OPC 服务器的进程中处理协议，则可略去在进程和多协议模式之间切换的附加执行时间。不过，OPC 客户端与 OPC 服务器之间的进程切换仍然存在。

利大于弊

使用高性能 DP、S7 和 SR OPC 服务器具有以下优点：

- 比使用多协议模式的性能更高。
- 简单组态。
- 通过 ProgID OPC.SimaticNET 进行访问。
- 多个客户端可同时使用服务器。
- OPC 服务器的稳定性不取决于客户端。

但是，也有一个缺点：

- 当使用高性能 DP、S7 或 SR OPC 服务器时，只能使用单协议模式。

如何才能激活高性能变量？

仅需选择组态程序"通信设置"(Communication settings) 中的 DP、S7 或 SR 协议，即可隐式激活这一功能强大的变量

3.5.2 自动化领域中 SIMATIC NET 的 OPC 服务器 - 如何使用它？

自动化领域中 OPC 服务器的潜在用途

该图显示了 SIMATIC NET OPC 服务器的广泛用途。

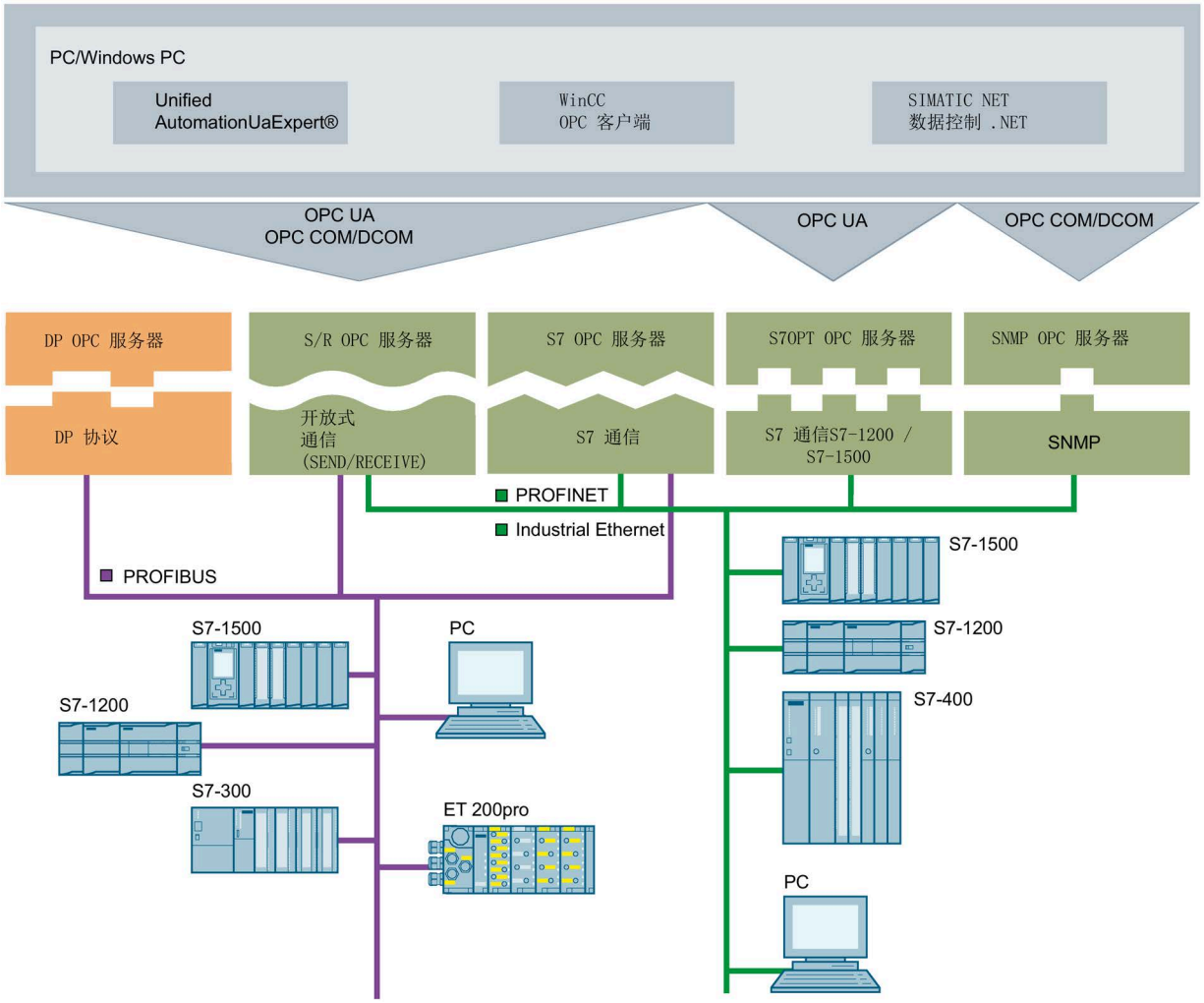

图 3-24 与 OPC 服务器的系统集成

3.5.3 SIMATIC NET 的 OPC 服务器 - 优点是什么？

OPC 服务器，优点显而易见

就 OPC 的性能而言，SIMATIC NET 提供各种常规优势，而且还涉及到对单个客户端进行编程和开发方面。不过，期间需要有调试人员的参与。

开放的 OPC 接口是 SIMATIC NET 的 PG/PC 上产品的中央接口。SIMATIC NET 的 OPC 服务器支持所有可供通信模块使用的通信协议和服务。

SIMATIC NET 的 OPC 服务器支持所有协议的 OPC 数据访问接口规范。对于具备传输事件机制的协议（S7 通信和 SNMP），还支持 OPC 报警和事件。

以下您将了解 SIMATIC NET 的 OPC 服务器是如何使用的，它具有哪些使用优势，以及适用属性有哪些。您还将看到如何通过 OPC 服务器实现过程数据的最佳访问。

SIMATIC NET 的 OPC 服务器允许访问工业通信网络 PROFIBUS 和 SIMATIC NET 的工业以太网。它使过程变量值可用于 OPC 客户端或发出来自伙伴设备的事件信号。为此，它可以借助 SIMATIC NET 通信处理器的协议软件通过通信网络访问伙伴设备。

调试的优点

- 仅使用一个与协议无关的接口；即，仅需为多个应用程序安装一个接口。
- 可对 SIMATIC NET 的通信网络进行简单访问。
- 可通过 SIMATIC NET 的通信网路，使用面向自动化工程中的大量应用的自动化系统。
- 可集成 Microsoft Office 产品。
- 使用 DCOM 时，安装于其它计算机上的应用程序也可通过全局网或局域网访问 OPC 服务器的功能。
- 例如，可以使用 SIMATIC Computing 在 Visual Basic 中快速创建简单的辅助程序。

程序开发的优点

- 使用与制造商无关的接口。这会在将来提供安全保证。您可为更大的市场提供服务并可重复使用您开发的产品。
- 已开发应用程序不依赖于某个制造商的通信系统，并可与差异最大的制造商的 OPC 服务器进行通信。
- 使用 OPC 接口，应用程序可对 OPC 服务器进行强大功能的访问，且基础通信网络可用。
- OPC 为 C 和 C++ 编程语言提供高性能接口。

- 用户无需熟悉协议和制造商特定的接口。
- 跟踪输出的选项将简化故障排除。
- 由于可仿真伙伴设备，因此无需安装额外设备即可开发程序。

OPC 服务器有哪些限制？

SIMATIC NET 的 OPC 服务器支持 OPC 数据访问与 OPC 报警和事件规范所需的全部接口。该服务器还提供最重要的可选接口，例如用于 OPC 数据访问的浏览接口。

可选接口将受到以下限制：

- 用于数据访问的 OPC 服务器不支持"OPC 公共组"。
- 用于数据访问的 OPC 服务器不允许写入时间戳和质量。
- 用于报警和事件的 OPC 服务器
  - 不能将工厂分为多个区域
  - 不能通过浏览来调查工厂区域

消息的类型和内容不由 OPC 规范规定。使用组态信息时，可指定报警是被指示为简单事件还是条件事件。

3.5.4 SIMATIC NET 的 OPC 服务器 - 有何作用？

SIMATIC NET 的 OPC 服务器用途

开放的 OPC 接口是 SIMATIC NET 的 PG/PC 上产品的中央接口。SIMATIC NET 的 OPC 服务器支持所有可供通信模块使用的通信协议和服务。

SIMATIC NET 的 OPC 服务器支持所有协议的 OPC 数据访问接口规范。对于具备传输事件机制的协议（S7 通信），还支持 OPC 报警和事件。

以下您将了解 SIMATIC NET 的 OPC 服务器是如何使用的，它具有哪些使用优势，以及适用属性有哪些。您还将看到如何通过 OPC 服务器实现过程数据的最佳访问。

SIMATIC NET 的 OPC 服务器允许访问工业通信网络 PROFIBUS 和 SIMATIC NET 的工业以太网。它使过程变量值可用于 OPC 客户端或来自伙伴设备的信号事件。为此，它可以借助 SIMATIC NET 通信处理器的协议软件通过通信网络访问伙伴设备（见图）。

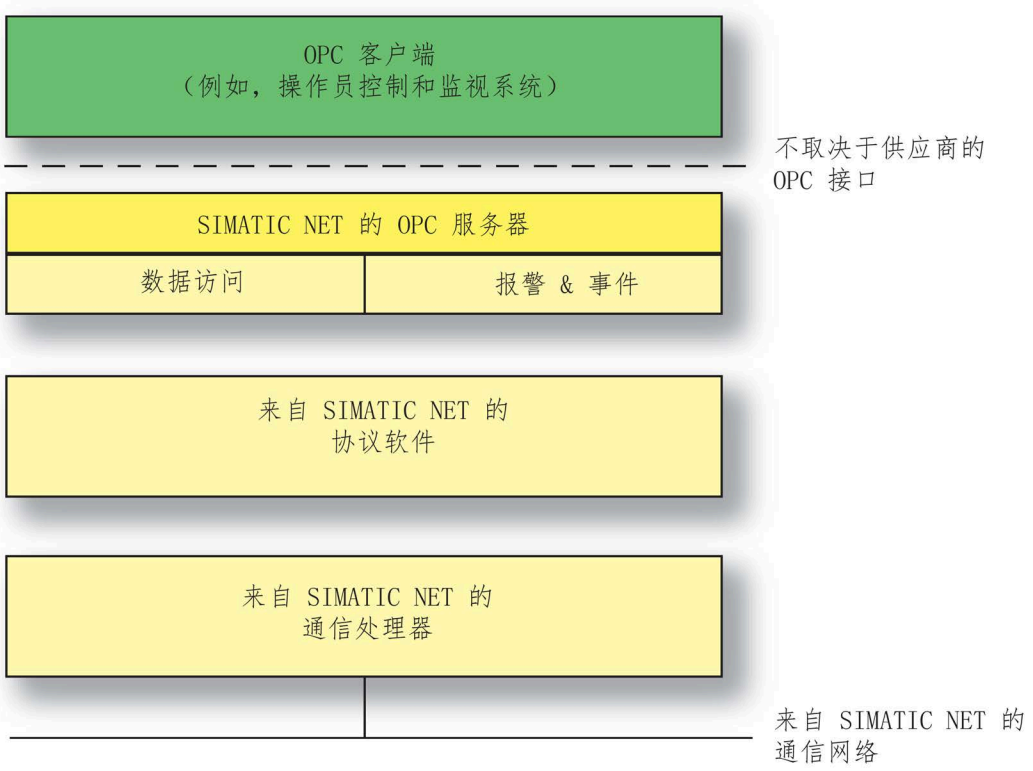

图 3-25　　具有 OPC 客户端的 SIMATIC NET 的 OPC 服务器

OPC 服务器提供哪些特殊服务？

由于 OPC 服务器可将任务分配到各种通信系统，所以 OPC 客户端可通过单个 OPC 服务器同时使用不同协议。如果在 OPC 服务器的组态中仅使用 1 个协议，则不必进行分配。这将优化数据吞吐量。

3.5.5 过程数据 - 如何实现最佳访问？

用于访问过程数据的不同机制

使用 OPC 数据访问，可访问不同类型的过程数据。 通过选择适当的方法，可影响应用程序的数据吞吐量。

使用某些协议，还可通过选择服务和在名称空间中构造变量来影响 OPC 服务器的性能。

选项数据吞吐量的任何提示？

以下部分中的信息：

- 使用组操作
- 访问 OPC 缓存
- 构造项
- 使用缓冲区服务

将有助于实现最大数据吞吐量

哪些方法最适合？

有不同的变量访问方式。 因为这些方式具有不同的特征，所以检查下文所描述的优缺点，然后选择最适合您的情况的访问选项。

本部分

- 方法 - 如何使用适合的方法？

将有助于做出正确的决定。

3.5.6 组操作 - 如何使用它们？

这是组操作的使用方式

使用多种方法，可将一个字段中的多个过程变量作为一个函数调用中的参数进行传输。使用组操作将有益于数据吞吐量，因为 OPC 客户端与 OPC 服务器之间存在更少的函数调用和过程切换。 这样一来，OPC 服务器便可通过将各个作业组合在一起，来自行优化网络上的通信。

如果通过 DCOM 上的远程服务器进行操作，则使用组操作将变得尤为重要，因为在此种情况下，函数调用会在网络上进行传输。

3.5.7 OPC 缓存 - 它是什么？

这是 OPC 缓存

OPC 缓存是 OPC 服务器的内部缓冲区，其中存储 OPC 项的最后获得值。

OPC 缓存 - 工作原理是什么？

OPC 服务器更新在激活组中所插入的全部激活项，并存储缓存中的读取值。 仅当成功地读取了项之后，缓存中的值才会有效。

从缓存中读取变量的速度要比访问设备更快。 假设缓存中的值对于特定应用程序更新得足够快，您应访问缓存。

缓存中的更新率由 RevisedUpdateRate 参数指定。

3.5.8 MaxAge - 它是什么？

MaxAge - 它是什么？

使用 OPC 数据访问 3.00 时，可更加详细地指定从缓存进行读取和采样的条件。

MaxAge 是采样前某个值的所需的最大老化时间（毫秒）。 如果读取作业未超出此老化时间，则会从缓存返回该值。 如果超出了时间，则必须再次从设备对其进行采样。

如果 MaxAge 值为 0，则意味着会对设备 (OPC_DS_DEVICE) 进行采样；如果 MaxAge 值为 1 到 0xFFFFFFFF（49.7 天），则意味着会从缓存 (OPC_DS_CACHE) 进行读取。

3.5.9 服务使用缓存 - 工作原理是什么（示例）？

使用缓存的示例

在此处，可看到与缓存一起使用的服务的示例：

IOPCSyncIO::Read(...,OPC_DS_CACHE,...)

从缓存同时读取多个 OPC 项的值、时间戳和质量。

IOPCAsyncIO2::Refresh(...,OPC_DS_CACHE,...)

在所有激活的 OPC 项的 OPC 客户端中生成回调，而不管该值为何。 将缓存中所存储的值发送到客户端作为当前值。

IOPCAsyncIO3::ReadMaxAge(...,MaxAge=500,...)

在组的所有 OPC 项的 OPC 客户端中生成回调，而不管该值为何。 如果数据不大于 500 毫秒，则会将缓存中的值（假设其存在）发送到客户端作为当前值。

3.5.10 协议 - 可优化哪些内容？

可对以下协议进行优化

SIMATIC NET 的 OPC 服务器为以下协议的变量服务提供优化算法：

1. S7 协议
2. 通过工业以太网进行的开放式通信服务 (SEND/RECEIVE)

将各变量的多个同时访问作业在内部转换为对伙伴设备的单次访问。 这可减少通过网络传输的数据包量，提高各包的利用率，并增加包的有效负载数据的比例。

此优化适用于读取和写入访问，并被作为默认值激活。 此优化算法对 OPC 客户端不可见。

哪些规则适应用于名称空间中 OPC 项的排列？

要允许优化，必须根据以下规则对名称空间中的 OPC 项进行排列：

- 同时读取或监视的 OPC 项**应**在伙伴设备的名称空间中进行连续排列。
  处理相关部分之间的较小间隙，但减少数据吞吐量。
- 同时写入的 OPC 项**必须**在名称空间中进行连续排列。
  为获得最佳的写入访问，不得存在任何间隙。 被作为优化结果创建的整个字段将传输至伙伴设备。 系统会使用未定义的值来覆盖间隙的地址区。 要避免此种情况，OPC 服务器需要再一次发送没有优化的各个访问作业。

3.5.11 缓冲区发送/接收服务 - 为何使用它们？

使用缓冲区发送/接收服务

要允许传输大型数据包，S7 通信、通过工业以太网进行的开放式通信服务 (SEND/RECEIVE) 和 PROFIBUS 应提供缓冲区发送/接收服务。 此时，将在通信伙伴之间发送数据包。 在伙伴显式传输发送作业时，数据的传输会对网络施加负载。

使用 SIMATIC NET 的 OPC 服务器时，可构造数据块。 这样一来，数据包的各部分便可被分配给 OPC 项。

3.5.12 缓冲区发送/接收服务 - 如何使用它们（示例）？

使用缓冲区发送/接收服务的示例

下图显示了 S7-400 设备如何将数据包发送至具有 S7 OPC 服务器的 PC 站。

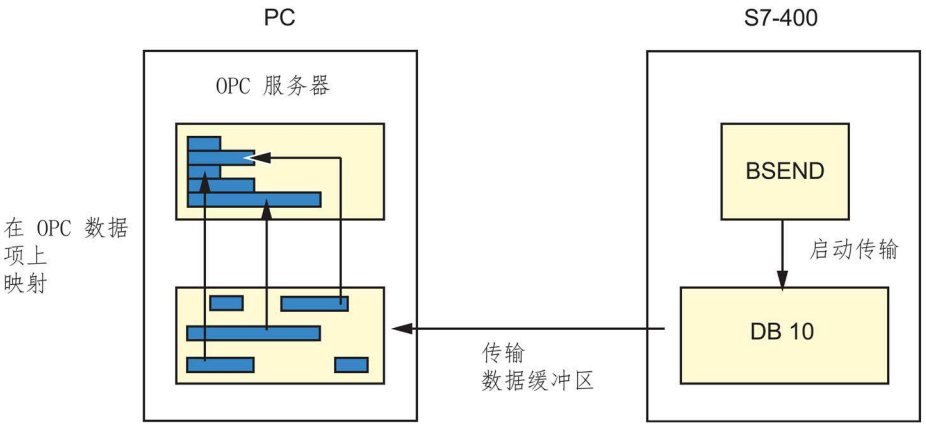

图 3-26 发送数据包

要使 OPC 服务器能够接收数据，可在激活组中插入 BRCV 类型的一个或多个 OPC 项。

在 S7-400 设备中，用户程序触发 BSEND 函数块。 BSEND 开始将整个数据区作为 PC 的缓冲区进行传输。

在 PC 上，将所接收的数据传输到 OPC 服务器。 现在，OPC 服务器即会将数据块的子区域映射到对应的 OPC 项。 如果监视这些 OPC 项，则在值已发生更改时，OPC 服务器会向 OPC 客户端发送回调。

3.5.13 方法 - 如何使用适合的方法？

3.5.13.1 同步访问 - 存在哪些类型？

同步访问的选项

对于 OPC 数据访问，存在两种数据访问方式：

同步读取和写入

读取和写入访问均可同步。

程序会启动同步函数调用来访问过程数据。 在此函数执行时，OPC 服务器将处理网络上的整个通信。 系统欠款将所读取的值传输到带有函数的返回参数的用户程序，随即此程序可继续进行带有下一批指令的序列。

同步访问的应用领域

当用户程序的较长中断不表示重大问题时，应始终使用同步访问。 同步访问始终是对伙伴设备数据进行的尽可能快的访问。

同步访问的优缺点

优点：

- 简单编程
- 高数据吞吐量，因为 OPC 客户端和 OPC 服务器之间的作业仅存在一个过程切换。

缺点：

- 应用程序将会中断，直到同步作业已完成。 仅当所有数据都已读取后，应用程序才能继续。 例如，如果在其自身的线程中未调用函数，则会在函数调用期间阻塞交互应用程序的用户界面。

3.5.13.2 异步访问 - 存在哪些类型？

异步访问选项

对于 OPC 数据访问，可通过以下方法访问数据：

异步读写

可以进行异步读或写访问。

程序会发送异步函数调用来访问过程数据。程序会立即接收到作业是否成功传输到 OPC 服务器的反馈。然后程序继续运行。

OPC 服务器已将 TransactionID 分配给作业，客户端随后可通过其识别该作业。

在以后的某一未定义时间，OPC 服务器调用 OPC 客户端的函数 (AsyncReadComplete) 或 (AsyncWriteComplete)。先前函数调用（读取或写入）的结果和 TransactionID 作为调用参数被传送到客户端。

通过网络传输数据的时间不取决于 OPC 客户端上运行的程序。

异步访问的应用领域

如果大量数据需要读取并且应用程序在处理作业时必须能够响应，异步访问将非常有用。

对 OPC 服务器缓存的异步访问无实际用途。
仅仅由于 OPC 客户端和 OPC 服务器之间的过程切换，就会造成处理器高负载情形。

同步访问的优缺点

优点：

因为实际通信与应用程序并行进行，所以应用程序仅会短暂中断。

缺点：

- 创建应用程序不是如此简单。必须在能够随时接收作业处理结果的应用程序中执行回调机制。
  Windows 程序具有作为标准的异步机制，因此其能够对用户输入做出响应。
- 当在一个作业中传送几个变量时，调用和回调时的过程跳转会造成较大负载。这一负载是同步访问时的两倍大小。

3.5.13.3 监视变量 - 此时会发生什么情况？

监视变量

监视变量时，OPC 服务器继续检查变量的值或质量是否已发生变化。

为此，OPC 客户端会将激活的 OPC 项添加到一个组并将其激活。然后，监视所有激活组中所有处于激活状态的 OPC 项。

OPC 客户端提供 OnDataChange() 函数。当值已更改时，OPC 服务器会调用此函数。作为参数，OPC 服务器传输 OPC 项的已更改的值、质量和时间戳。

OPC 客户端将因监视变量而受到任何负载。仅当检测到更改时，客户端的程序才会运行。

要在过程变量快速变化时防止 OPC 客户端承载过多的更改消息，可在使用组特定参数 RequestedUpdateRate 和 RevisedUpdateRate 调用的程序中指定最小更新率。

受噪声影响的模拟值会导致更改消息的快速序列，因为该值始终发生轻微的更改。使用 DefaultGroupDeadBand 参数，百分数为未发出更改信号的组的所有项定义范围。该范围的绝对大小是已组态上限与下限之间的差别的百分比。

仅在"符号编辑器"中定义了变量的值范围时，才会出现这种情况。

何时应监视变量？

在程序始终需要过程或过程的部分的最新数据时，监视变量是最佳的解决方案。

监视变量的优缺点

优点：

- 仅当过程数据已发生更改时，才会通知应用程序。这将降低 CPU 利用率。
- 数据吞吐量非常高，因为过程切换比较少。根据项结构的组成，可实现良好的优化。
- 客户端可启用或禁用项特定或组特定的变量监视。
- Windows 程序将异步机制作为标准，以便它们可对用户输入做出响应。

缺点

- 从更改过程中的值到将新值传输至客户端的响应时间比组的更新率高。
- 创建应用程序不是如此简单。应用程序需要异步部分来接收值更改。

更新率 (RevisedUpdateRate)

可通过用户程序 (RequestedUpdateRate/RevisedUpdateRate) 设置的"更新率"参数将指定用于检查激活的 OPC 组中 OPC 项的值的最短时间间隔。

服务器检查值是否已更改。如果存在新值，则服务器向客户端报告新值。发送客户端的消息的速度不如客户端所设置的"RevisedUpdateRate"快。如果值更改快于更新率中指定的时间间隔，则不会将中间值告知客户端！

扫描周期

扫描周期（以 ms 为单位）指定 OPC 服务器使用新通信作业更新 OPC 项的值的频率。

扫描周期和更新率之间的关系

SIMATIC NET 的 OPC 服务器所使用的更新率 (RevisedUpdateRate) 是在组态期间指定的扫描周期的倍数。最小更新率与扫描周期相同。

协议特定的扫描周期之间的关系

由于 SIMATIC NET OPC 服务器可同时使用不同协议的变量，因此 OPC 服务器的最小更新率是为激活协议的扫描周期设置的最低值。

3.5.14 百分比死区 - 如何使用此参数？

这是百分比死区的使用方式

“百分比死区”参数定义项的范围，在该范围中不报告值的更改。 该范围的绝对大小是已组态上限与下限之间的差别的百分比。

在下例中假定以下条件： 百分比死区 = 10% = 1 个单位（其中上限 = 10，下限 = 0）。

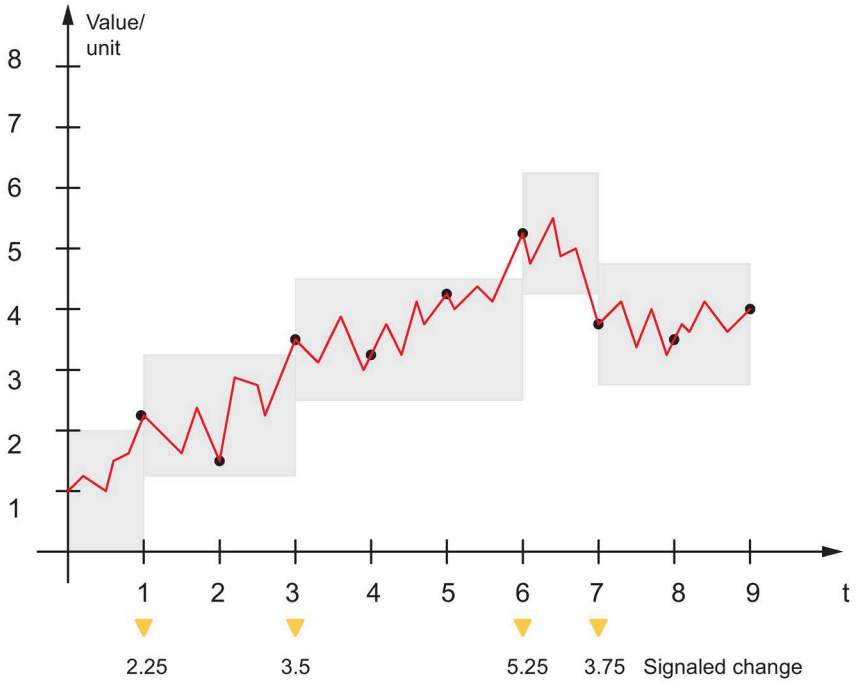

图 3-27 不发出更改信号的范围

在向客户端发出每个值更改信号后，围绕最后发出信号的值设置一个区。 仅当值在读取访问期间值离开上一个区（在监视值的情况下）时，才会将更改消息发送到客户端并指定一个新区。

通过使用 OPC 数据访问版本 3.00 所引入的接口函数，可在组内为特定项设置“百分比死区”参数。

3.5.15 采样速率 - 如何将其用于特定项？

设置数据访问规范的 3.00 及以上版本的采样速率

使用“RequestedUpdateRate”和“RevisedUpdateRate”设置用于监视变量的更新率。 更新率适用于一个或多个激活的 OPC 组。

除此之外，自数据访问规范 3.00 起，可使用可选的“IOPCItemSamplingMgt”接口设置采样速率。 它们可高于或低于组更新率 (RequestedUpdateRate/RevisedUpdateRate)。 这允许调整 OPC 组中各项的采样速率以更准确地适应更改的实际速率。 在此种情况下，必须在 OPC 服务器上缓冲项特定的值。

OPC 组中采样速率的示例

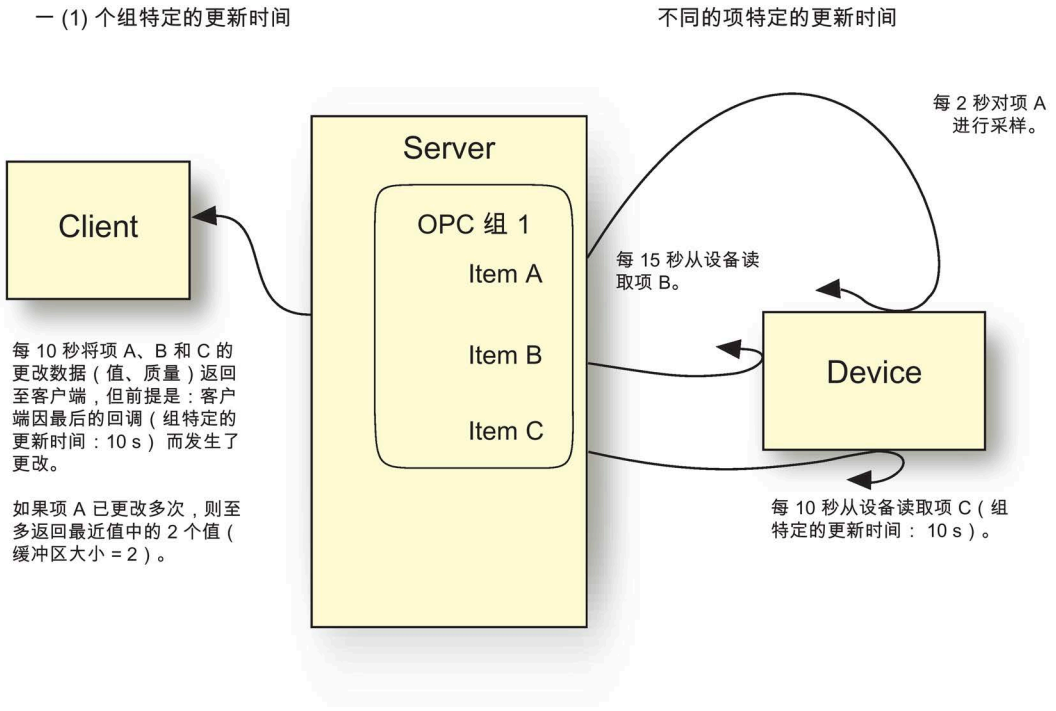

图 3-28 OPC 组中的采样速率

服务器上的 OPC 组 1 具有 A、B 和 C 这三个项。组更新率为 10 秒。 组和所有项均将处于激活状态。 按照如下方式对这些项进行组态：

- 项 A
  - 采样速率：2 s
  - 缓冲：激活
  - 缓冲区大小： 2
- 项 B
  - 采样速率：15 s
- 项 C
  - 无已组态的采样速率

在使用 10 秒的组更新率情况下，通过使用回调函数将带有时间戳的项 A、B 和 C 的更改值（值、质量）传输到 OPC 客户端。

如果组更新率 (10 s) 内存在项 A 的多个已更改数据，则会将多达 2 个的最新更改值传输到 OPC 客户端（缓冲区大小 = 2）。

OPC 客户端的组更新率不由采样进行更改。 如果项的所需采样速率和组的更新率相同，则与规范版本 2.xx 中的响应相比没有任何区别。

缓冲和传输项

以下规则当前适用于具有组更新的项的返回：

1. 如果质量因最近更新而发生了更改，则将该项返回至客户端。
2. 如果值因最近更新而发生了更改并且总更改超过死区（如果适用），则将该项返回至客户端。

如果采样速率短于组更新率，则服务器需要额外的逻辑来决定在下一次更新中被返回至客户端的内容。

如果未启用缓冲且至少一个采样符合上述条件 1 和 2，则会将最新值返回至客户端。

此外，当最新值不符合条件 1 和 2 时，还会将最新采样返回。

如果启用了缓冲，则服务器在其读取符合条件 1 和 2 的变量之前，不会启动缓冲采样。当服务器开始为项缓冲采样后，如果新采样的质量或值与先前采样相比有所不同，则会将新采样添加到缓冲区。

如果新采样和缓冲区中的最后采样相同，则服务器将仅更新缓冲区中最后采样的时间戳。

扼要重述： 已返回至客户端的采样集将是一系列值，而这些值都不同于先前的采样，并且具有反映该项被知道具有该值的上次时间的时间戳。

如果项具有使用特定 OnDataChange 回调返回的多个值/质量/时间戳，则将存在多个 ClientHandle，并根据集合的大小，使用对应的值/质量/时间戳（这三个构成一组）进行返回。

通过 OPC 数据访问 3.00 引入的用于项缓冲的函数称为“SetItemBufferEnable”。

OPC 服务器向客户端发出值信号时，需要使用 OnDataChange() 函数。

参考

4

查找 SIMATIC NET 文档

- 西门子网上商城
  可以在西门子网上商城中找到 Siemens 相关产品的订货号：
  (https://eb.automation.siemens.com/goos/WelcomePage.aspx?regionUrl=/&nodeID=1000000&view=intranet&infoTypeID=*&language=en#topAnch)
- Internet 上的文档
  在 Siemens 自动化客户支持 Internet 页面上可找到 SIMATIC NET 手册：客户支持链接 (https://support.automation.siemens.com/WW/view/zh)
  转到所需产品组并进行以下设置："条目列表"选项卡，条目类型"手册/操作说明"
- STEP 7 安装文档
  可以通过开始菜单"开始 > 所有程序 > Siemens Automation > 文档"(Start > All Programs > Siemens Automation > Documentation) 找到 PG/PC 上 STEP 7 产品在线文 档中包含的手册。
- "SIMATIC NET PC 软件"DVD 上的文档
  可在"SIMATIC NET PC 软件"DVD 上的以下文件夹目录中找到有关"SIMATIC NET PC 软件"的所有文档："%ProgramFiles%\Siemens\SIMATIC.NET\doc"。

参见

文档的链接：
(https://www.automation.siemens.com/simatic/portal/html_00/techdoku.htm)