第 5 章

全光自动驾驶网络关键技术

数字化转型对运营商网络及服务能力提出新的挑战。全光自动驾驶网络综合运用人工智能算法、大数据、算力、自动化协议等关键技术，使能新一代智慧全光组网解决方案，让网络规划更精准、部署更快捷、管控更简单、运维更高效。同时变被动响应为主动运维，进一步提升光网络的灵活性和可靠性。

5.1 AI 算法技术

算法是实现人工智能（Artificial Intelligence，AI）的基础。借助 AI 算法对海量数据进行特征提取，结合专家经验从已有的数据特征中建立模型，从而快速解决已知问题。在全光网络中引入 AI 技术，关键在于根据不同的场景选取合适的 AI 算法，并有针对性地加以改进和适配。常用的 AI 算法包括神经网络算法、时间序列预测算法、聚类算法、逻辑回归算法等。在不同场景下考虑到解决实际问题的需要，可组合使用上述各种典型算法。

5.1.1 AI 和机器学习

对于什么是智能，业界仍然存在争议。目前智能研究的唯一对象还是人类，尽管人类对自身智能的认知历史很长，从哲学、科学角度都进行了探索，但所知有限，无论对智能是如何产生的，还是对构成智能的各种要素都知之甚少。

人工智能亦称智械、机器智能，其含义是由人制造出来的机器所表现出来的智能，即通过普通计算机程序来呈现人类智能的技术。计算机科学发展加快了 AI 领域研究的步伐，这些工作主要集中在寻求如何使计算机以更加智能化的方式运算。

AI 的核心问题是如何构建出能够与人这一智能体类似的推理、知识、规划、学习、

交流、感知、移物、使用工具和操控机械的能力等，达到甚至远超人类的精准表现是 AI 领域发展的长远目标。

目前，弱人工智能已经取得初步成果，围绕影像识别、语言分析、棋类游戏等单方面的 AI 能力达到了超越人类的智能化水平。AI 的通用性意味着解决上述问题是一样的 AI 程序，无须重新开发算法即可直接使用现有的 AI 完成任务。达到具备思考能力的强人工智能还需要时间研究，比较流行的方法包括统计方法、计算智能和传统意义的 AI。

利用 AI 技术辅助和增强人类自身的智能，成为必然的方法和途径。已有大量的工具应用了人工智能，其中包括搜索、数学优化和逻辑推演。基于仿生学、认知心理学，以及基于概率论和经济学的算法等也在逐步探索当中。

AI 的发展呈现出一条从以“推理”为重点，到以“知识”为重点，再到以“学习”为重点的自然、清晰的脉络。作为 AI 研究分支之一，机器学习在近 30 年已发展成为一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、计算复杂性理论等多门学科。机器学习理论主要是设计和分析一些让计算机可以自动“学习”的算法，并以此为手段解决人工智能中的问题。机器学习算法是一类从数据中自动分析获得规律，并利用规律对未知数据进行预测的算法。因为学习算法中涉及了大量的统计学知识，与推断统计学的联系尤为密切，机器学习理论也被称为统计学习理论。机器学习主要关注能够实现的、行之有效的学习算法，然而很多推论问题属于无程序可循的情况，此类研究集中在开发相对容易处理的近似算法。

机器学习可以分为以下类别。

1. 监督学习

监督学习是从给定的训练数据集中通过学习得到一个函数（或称为模型），当新的数据到来时可以根据此函数预测结果。监督学习的训练集要求包括输入和输出，也可以说是特征和目标。训练集中的目标是由人标注的，也就是说人通过数据将先验的知识表达出来，机器学习的内容就是人标注在数据集中的先验知识。监督学习过程分为训练过程和推理过程。

（1）训练过程：类似人类学习的过程，如图 5-1 所示，通过不断地对数据集进行迭代计算，调整函数（模型）的参数，使得到的模型能够更准确地描述数据集传递的先验知识。

（2）推理过程：类似人类工作的过程，如图 5-2 所示，通过模型对输入的数据进行

计算，输出结果。在这个过程中，模型的参数不发生变化。

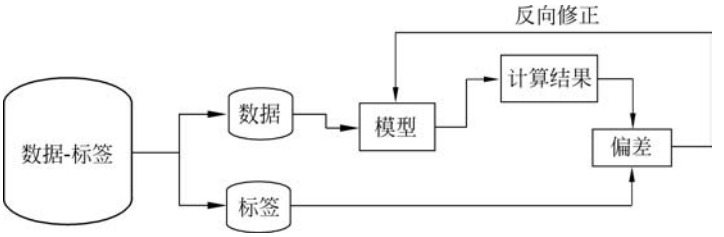

图 5-1　监督学习的训练过程

图 5-2　监督学习的推理过程

这种训练和推理分离的机器学习方式，也称为离线学习。与之相对的概念是在线学习，后者通过在线地进行训练和推理，具备更强的场景适应性和泛化能力。

2. 无监督学习

无监督学习与监督学习相比，训练集没有人为标注的结果。如图 5-3 所示，常见的无监督学习算法包括生成对抗网络（Generative Adversarial Network，GAN）和聚类。

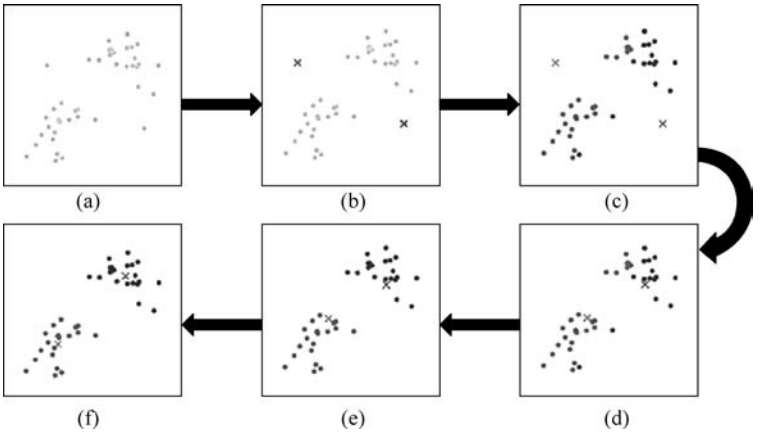

图 5-3　基于聚类的无监督学习过程

聚类算法是最简单的无监督学习方法，分为基于距离的聚类 k-means、基于密度的

聚类 DBSCAN 等。这类算法的特点是不需要进行样本标注，通过距离或者密度的评价标准就可以将样本进行分类。

无监督学习方法一般用于数据的预处理分析，可以将大量数据进行初步的分类。一般对于相同的问题，无监督学习的效果不如监督学习。但在实际业务场景中，大量数据往往是没有标签的，因此进行聚类等无监督学习是非常有效的数据预处理手段。

3. 半监督学习

实际应用过程中，如果少量数据或者部分数据有标签，而其余的数据没有标签，那么此类场景适合使用半监督学习方法。

半监督学习介于监督学习与无监督学习之间，对于部分样本进行标注，而非全部数据都进行标注。

比如同一个问题有两个数据集，一个数据集有标签，另一个数据集没有标签。Label＝{(xi，yi)}，Unlabeled＝{(xi)}，并且数量上 Label＜＜Unlabeled。如果使用 Label 数据集，对应的是监督学习；如果仅使用 Unlabeled 数据集，则是无监督学习。

少量的有标签数据获得的决策边界往往不是真实的分类边界，因此通过加入无标签数据，将分类边界进行调整，可以更好地拟合实际的分布。

4. 主动学习

主动学习是解决部分数据标注问题的另外一种思路。

首先解释什么是样本信息。样本信息指在训练数据集中每个样本带给模型训练的信息。由于不同样本的样本信息是不同的，即每个样本对模型训练的贡献有大有小，它们之间存在差异。

因此，为了尽可能地减小训练集及标注成本，在机器学习领域中，提出主动学习(Active Learning)方法，优化分类模型。

在某些场景下有类标的数据比较稀少，而没有类标的数据相当丰富，对数据进行人工标注需要付出很大的代价。如图 5-4 所示，学习算法可以主动地提出一些标注请求，将一些经过筛选的数据提交给专家进行标注。

主动学习的模型为

$$A=(C,Q,S,L,U) \tag{5-1}$$

式中，C———一组或者一个分类器；L——用于训练已标注的样本；Q——查询函数，用于从未标注样本池 U 中查询信息量大的信息；S——督导者，可以为 U 中样本标注正

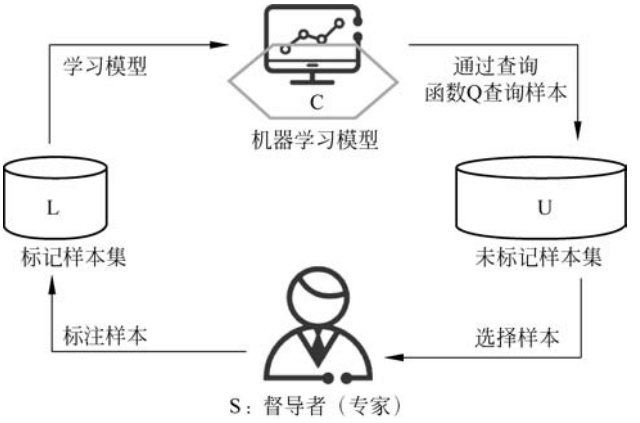

图 5-4　主动学习筛选数据的过程

确的标签。学习者通过少量初始标记样本 L 开始学习，通过一定的查询函数 Q 选择出一个或一批最有用的样本，并向督导者询问标签，然后利用获得的新知识来训练分类器和进行下一轮查询。主动学习是一个循环的过程，直至达到某一停止准则为止。

查询函数 Q 用于查询一个或一批最有用的样本。那么，什么类型的样本属于有用的样本呢？或者说查询函数查询的是什么类型的样本呢？在各种主动学习方法中，查询函数设计最常用的策略是不确定性准则（Uncertainty）和差异性准则（Diversity）。

对于不确定性，可以借助信息熵的概念加以理解。信息熵既可以衡量信息量，又可以衡量不确定性。信息熵越大，代表着不确定性越大，所包含的信息量也就越丰富。事实上，有些基于不确定性的主动学习查询函数使用了信息熵进行设计，如熵值装袋查询（Entropy Query-by-bagging）。所以，不确定性策略就是要想方设法地找出不确定性高的样本，因为这些样本所包含的丰富信息量，对我们训练模型来说就是有用的。

差异性又该如何理解呢？前面提到查询函数每次迭代中查询一个或者一批样本。我们自然希望查询样本所提供的信息是全面的，各个样本包含的信息不重复、不冗余，即样本之间具有一定的差异性。在每一轮迭代抽取单个信息量最大的样本加入训练集的情况下，模型通过反复训练，以新获得的知识参与对样本不确定性的评估能够有效地避免数据冗余。如果每次迭代查询一批样本，那么需要采取措施保证样本之间的差异性，避免数据冗余。

部分观点认为主动学习也属于半监督学习的范畴，但实际上两者并不相同。半监督学习和直推学习（Transductive Learning）以及主动学习，都属于利用未标记数据的学习技术，然而它们的基本思想还是有区别的。常见的主动学习方法如下所述。

（1）基于不确定度缩减的方法。

这类方法选择那些当前基准分类器最不能确定其分类的样例进行标注。由于以信息熵作为衡量样例所含信息量大小的度量，而信息熵最大的样例对应着上述最不能确定其分类的样例。从几何角度分析，基于不确定度缩减的方法优先选择靠近分类边界的样例。

（2）基于版本缩减的方法。

这类方法选择那些训练后能够最大程度缩减版本空间的样例进行标注。在二值分类问题中，基于版本缩减方法所选择的样例总是差不多平分版本空间。

基于委员会投票选择（Query By Committee，QBC）算法从版本空间中随机选择若干假设构成一个委员会，然后选择委员会中的假设预测分歧最大的样例进行标注。采用 Bagging、AdaBoost 等分类器集成算法从版本空间中产生委员会，能够优化其构成。

（3）基于泛化误差缩减的方法。

这类方法试图选择那些能够使未来泛化误差最大程度减小的样例。对应过程为：首先选择一个损失函数用于估计未来错误率，然后将未标注样例集中的每一个样例都分别估计可能给基准分类器带来的误差缩减，选择估计值最大的那个样例进行标注。

该方法直接针对分类器性能的最终评价指标进行学习，所需计算量较大，同时损失函数的精度会对性能产生较为严重的影响。

5. 强化学习

强化学习是指机器为了达成目标，随着环境的变动，会逐步调整其行为，并评估每一个行为之后所得到的回馈是正向的还是负向的。传统机器学习与强化学习的对比如图 5-5 所示，强化学习原理示意图如图 5-6 所示。

图 5-5　传统机器学习与强化学习的对比

强化学习的关键是抽象出合理的期望（Reward）函数。比如棋类游戏，可以通过胜

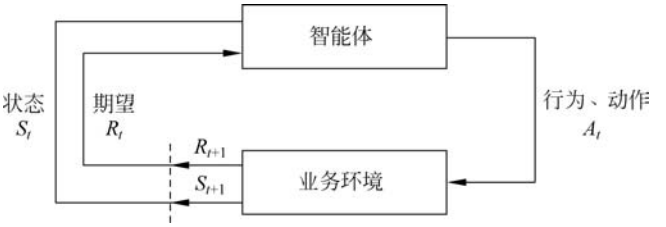

图 5-6　强化学习原理示意图

负概率作为期望函数。实际系统中，获得期望函数往往比较困难，很多决策难以快速给出反馈的结果。

5.1.2　机器学习的常见算法

下面介绍几种典型的 AI 算法。

（1）神经网络算法：神经网络（Neural Network）算法是一类仿生学算法，其特点是构造了一组相互连接的人工神经元（节点），每个神经元能够响应多个输入。深度神经网络通过构造若干隐藏层，可以进行复杂的非线性拟合。该算法可广泛用于目标识别、自动分类、非线性预测等领域。

（2）时间序列预测算法：也称为自回归移动平均模型（Autoregressive Integrated Moving Average Model，ARIMA）算法。ARIMA 属于传统 AI 类型的数据分析方法，通常被用在基于时间序列的平稳随机过程的数据预测。

（3）聚类算法：聚类（Clustering）算法属于无监督的机器学习算法，可自动根据数据的相似度、相关性进行分组分类。该算法可用于实现模式识别（分类）、根因分析等领域。

（4）逻辑回归算法：逻辑回归（Logistic Regression）是一种非线性回归模型，其特征数据可以是连续变量，也可以是分类变量或哑变量。逻辑回归属于机器学习方法，常用于估计某种事物的可能性。

在不同场景下，可能会组合使用上述各种算法，有针对性地解决具体实际问题。同时，在机器学习类算法中训练数据非常重要，成功的应用不仅仅依赖于算法本身，获取大数据也是关键因素。例如：为解决预测性问题，需要组合多种算法进行特征提取、异常数据处理、趋势拟合等适配处理，并且使用现网环境数据进行反复调整和验证，才能构建最佳应用。

在光网络中采用的 AI 算法，实现的关键在于根据不同的场景选取合适的算法，并

针对性地进行改进和适配。

5.1.3 AI 算法典型应用

在全光自动驾驶网络中，由于光层是专业性强、复杂度高的多参量系统，其运维问题成为主要难点之一。通过引入 AI 算法，可以提升光层运维自动化程度，如图 5-7 所示。

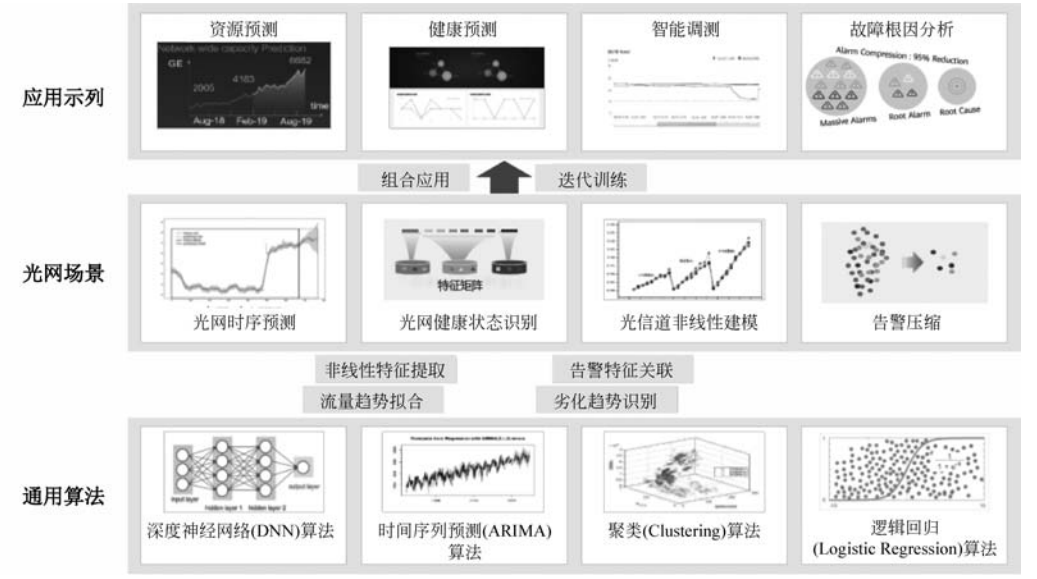

图 5-7　AI 算法在光网络的应用场景和示例

（1）将回归和时间序列预测算法应用在光网时序预测场景，对波长、端口、单板等使用情况进行资源预测，根据预测提前规划时序等资源，面向新业务的发放需求实现资源快速开通。

（2）将时间序列算法应用在光网健康状态识别场景，对光层性能实时监视采集，对光性能数据进行健康预测，存在健康风险时进行主动预警。

（3）将聚类算法和逻辑回归算法应用在光网告警压缩场景，对海量告警进行压缩和根因分析，提升故障处理效率。

（4）将深度神经网络等算法应用在光信道非线性建模场景，通过对光层性能的建模分析与仿真预测，寻找最优的光层功率智能调测方式，使光路获得最理想的传输效果。

AI 技术在光网络中的应用仍处于探索当中，下面以故障智能预测类应用为例，探讨 AI 技术实现网络智能化的基本处理框架。如图 5-8 所示，在整个流程中算法、大数据和算力三要素相互配合，缺一不可。

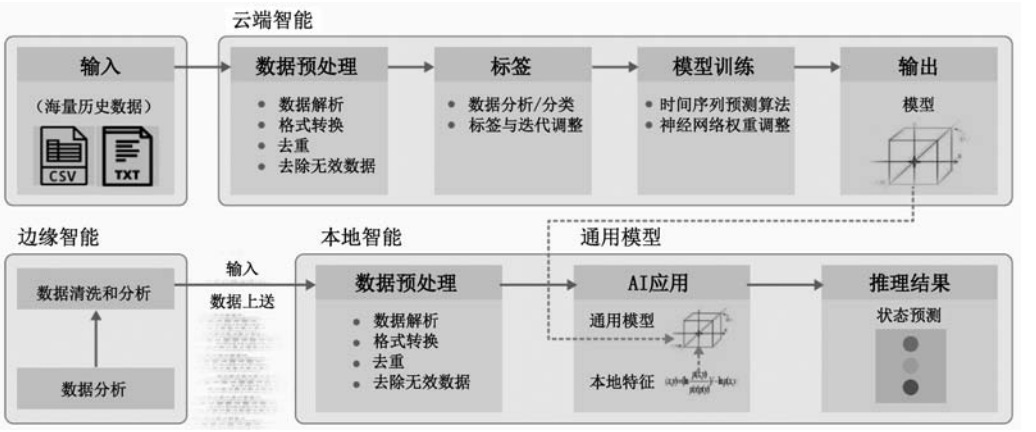

图 5-8　故障智能预测类应用示意图

涉及的关键过程如下所述。

（1）云端智能利用海量数据进行模型训练。

（2）边缘智能设备实现秒级数据采集和实时数据清洗及分析。

（3）本地智能应用进行数据特性提取，将数据导入通用模型或本地特征化的改进模型进行在线推理，得出状态预测结果，甚至给出处理建议。

应用到的关键算法包括时间序列预测算法、特征提取算法、逻辑回归算法、神经网络算法。

5.2　大数据技术

全光自动驾驶网络中，大数据技术是为了映射物理网络状态，建立光层数字孪生（Digital Twins）。通过物理网络不同类型的光电探针单元（Optical/Electrical Sensor），采集光功率、光信噪比（Optical Signal-to-Noise Ratio，OSNR）、误码率（Bit Error Rate，BER）、光谱、偏振态（State of Polarization，SOP）、偏振模色散（Polarization

Mode Dispersion，PMD）、偏振相关损耗（Polarization-Dependent Loss，PDL）等物理特征参数，从而获取实时监控数据。引入AI算法技术对这些数据进行训练和建模，可以更好地实现上层应用，提升网络自动化程度，节省运营成本。

为了支撑光网智慧运维的智能监控、智能预测以及智能保障，需要对大数据的产生、采集到最后的存储构建完整的流程。

5.2.1 光电 Sensor

光电 Sensor 指设备上的探测和采集装备，能够获取数字化的光网络物理层参数，支撑全光自动驾驶网络进行实时网络监控、高性能传输、自动部署和自愈自优。为了全面监控光网络，可从 4 个层次构建高精度的光网 Sensor 体系。

1. 光部件层 Sensor

单板和器件是网络的基础构成单元，光部件层 Sensor 用来监视各个器件的工作状态，基于状态数据可以分析判断器件是否正常，是否存在劣化、失效的风险。例如对于光模块需要实时监视激光器的状态，一般而言光模块的输出功率是稳定的，但是随着运行时间的增加，受老化因素的影响激光器发光效率会降低，需要不断增加偏置电流以维持稳定的功率输出，因此，可以通过监控激光器的偏置电流变化来分析光模块的使用寿命。光部件层 Sensor 主要监控器件工作状态的电流、电压、温度、功率、频偏等，对于控制类单板还需要监视 CPU、内存、Flash 的工作状态。

2. 光链路层 Sensor

光链路代表了光节点之间的物理连接，是传输质量监控的基本单元。在光传送网（OTN）中，光链路对应光传输段（OTS）和光复用段（OMS），主要监控波分系统合波信号的状态。以光传输段为例，通过监视上游光放大器的输出功率和下游光放大器的输入功率，可以计算本段链路的光纤衰减。对光纤衰减进行长期监视，可以发现光纤是否劣化，及时做出调优或维修决策。如图 5-9 所示，为了全面掌握光链路的工作状态，光链路层 Sensor 还需要监视光放大器增益，用于评估光纤衰减是否得到精确补偿；监视在用波长，用于评估资源利用率；监视单波功率、OSNR，用于评估链路上波长之间的平坦度；监视光谱状态，用于评估滤波状态等。

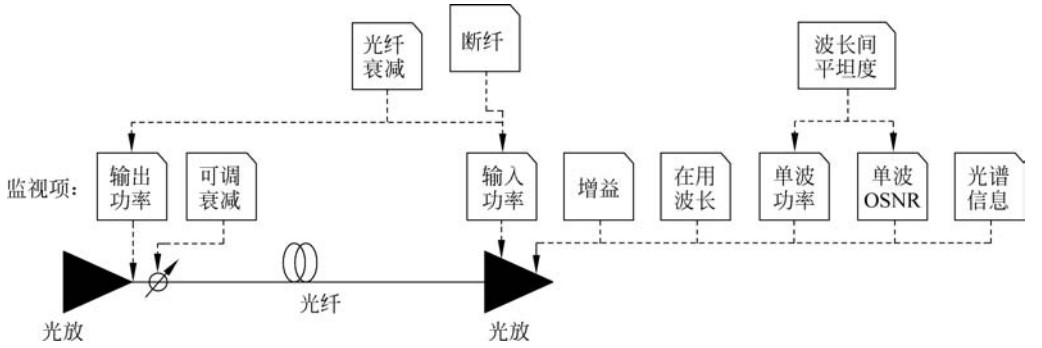

图 5-9　光链路层 Sensor 示意图

3. 光信道层 Sensor

光信道对应 OTN 网络中从发端光转换器单元(OTU)到收端 OTU 之间的单波传输通路，其收发两端 Sensor 可以监视光信道的端到端性能，包括光信道速率、调制码型、中心频率、频宽等基本信息，以及光信道的发送和接收光功率、OSNR、PMD、CD、SOP、BER 等。通过对收端性能(如 BER)的长期监视，可以分析光信道的余量状态和劣化趋势，用于及时进行性能调优，保障传输质量。如图 5-10 所示，光信道在中间传输过程中会经过一条或多条光链路，在光链路上监视光信道的状态信息，可以构建出全路径的实时性能分布，用于自动分析光信道的劣化根因和潜在风险，也可以用于调优仿真，即分析改变链路上的功率状态后，收端性能的变化结果。

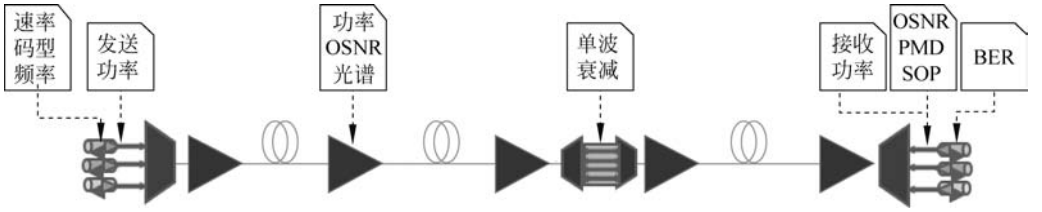

图 5-10　光信道层 Sensor 示意图

4. 光业务层 Sensor

光业务指光信道上承载的客户业务，为保障业务 SLA(Service Level Agreement)，光业务层 Sensor 主要监视流量、丢包率、误码率、时延等。

5.2.2 大数据采集

全光自动驾驶网络具有自动部署、自愈自优的特点，依赖网络监控能力的实时性要求。面向大规模的组网应用，需要监视的参数多、数量大，更加离不开高效率的大数据采集技术。

如图 5-11 所示，传统的网络监控通过 Pull 模式（一问一答模式）轮询获取设备监控数据（如接口流量），限制了接入的网络节点数目。虽然 SNMP 中的 Trap 机制是采用 Push 模式（推送模式），能够在设备产生告警和出现异常事件时及时推送数据，但是这些数据仅限于描述告警或者异常事件的通知，并不支持对实时功率值、流量大小等状态信息的采集。此外，基于 SNMP 等传统网管工具的数据采集频率只能达到分钟级，难以精确反映网络的实际状态。

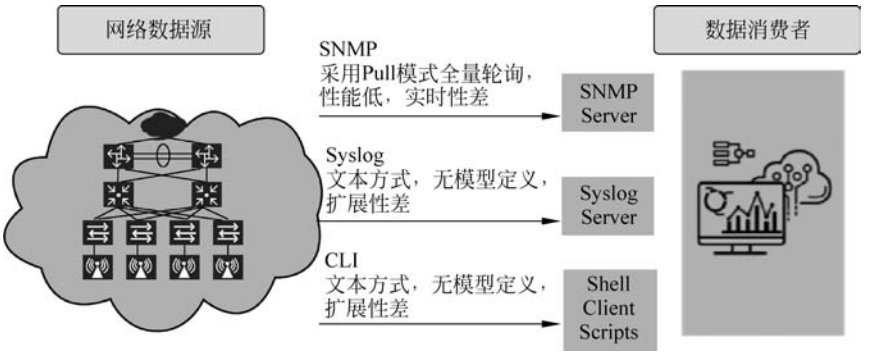

图 5-11 传统数据采集方式示意图

针对大规模、高性能网络的监控需求，Telemetry 数据采集技术应运而生，可根据网络各类功能平面的特点，提供不同的模型、编码、采集及传输协议，如表 5-1 所示。

表 5-1 新的网络数据采集框架

项　目	控 制 面	管 理 面	转 发 面	外 部 数 据
数据模型	YANG 自定义	MIB Syslog YANG 自定义	Template YANG 自定义	YANG
数据编码	GPB JSON XML Plain	GPB JSON XML	Plain	GPB JSON XML Plain

续表

项　　目	控　制　面	管　理　面	转　发　面	外 部 数 据
采集协议	gRPC NETCONF IPFIX Mirror	gRPC NETCONF	IPFIX Mirror	gRPC
传输协议	HTTP TCP UDP	HTTP TCP	UDP	HTTP TCP UDP

相比传统的数据采集模式，Telemetry 技术提供 Push 模式的海量数据上报，支持毫秒级上报周期，能够有效满足智能运维的目标。如表 5-2 所示，全光自动驾驶网络的控制器通过 Telemetry 采集光网络中光电 Sensor 的监控数据，运用 AI 算法对获取到的数据进行分析、呈现，从而完成对网络的主动监控和异常预测，实现全网智能运维。

表 5-2　传统模式的数据采集和 Telemetry 模式对比

采集技术	传 统 采 集	Telemetry 采集
采集方式	Pull 采集，每次数据采集都需要网管发起查询	Push 采集，网管一次订阅，网元持续推送数据
采集模式	集中式采集，数据统一汇总至主控后再上传网管	分布式采集，主控板、线卡直接向网管发送数据
采集位置	软件采集，依靠 CPU 采集数据，采集间隔长	硬件采集，硬件芯片直接采集并上报数据，实现毫秒级采集
数据模型	非结构化数据，信息处理效率低	结构化数据，存储传输效率高

1. 数据模型

Telemetry 的关键技术流程为根据 YANG 模型描述的结构组织采集原始数据，使用 GPB(Google Protocol Buffer)编码格式和 gRPC(Google Remote Procedure Call)协议在加密后的通道上对这些数据进行传输，从而实现原始数据采集、数据模型、编码类型、传输协议的融合。

YANG 模型是一种模块化语言，其核心是把任何对象都以树的方式进行描述。在 YANG 模型定义中，数据的层次结构被模型化为一棵树，树有 4 种节点(容器、列表、叶子列表、叶子)，每个节点都有名称，要么有一个值，要么有一个子节点集。

```
YANG Example:
container system {
    container 1oqin {
        leaf messaqe {
            type strinq;
            description
                "Message given at
                start of login session";
        }
    }
}
```

YANG 提供了对节点清晰简明的描述以及节点间的交互关系。YANG 数据层次结构包含了列表的定义，其中列表条目由关键字识别并加以区分。

```
YANG Example:
list user {
    key "name";
    config true;
    description "This is a list of users in the system.";
    leaf name {
        type string;
    }
    leaf type {
        type string;
    }
    leaf full - name{
        type string;
    }
}
```

上述列表可定义为由用户排序，也可定义为系统自动排序。YANG 对由用户排序的列表定义了执行列表条目顺序调整的操作，如下面这段 RFC 7223 定义的接口 YANG 模型所示。

```
+-- rw interfaces
|     +-- rw interface* [name]
|       +-- rw name                              string
|       +-- rw description?                        string
|       +-- rw type                              identityref
|       +-- rw enabled?                           boolean
```

```
|      +-- rw link-up-down-trap-enable?      enumeration
+-- ro interfaces-state
   +-- ro interface* [name]
        +-- ro name                    string
        +-- ro type                    identityref
        +-- ro admin-status              enumeration
        +-- ro oper-status               enumeration
```

2. 数据编码

XML(Extensible Markup Language)指可扩展标记语言,被设计用来传输和存储数据,是各种应用程序之间进行数据传输最常用的编码方式。XML 提供了一套跨平台、跨网络、跨应用程序的语言的描述方式。使用 XML 可以方便地实现数据交换、系统配置、内容管理等。

JSON(JavaScript Object Notation,JS 对象表示法)是一种轻量级的数据交换格式。它基于 ECMAScript 的一个子集,采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言,它易于人的阅读和编写,同时也易于机器解析和生成,并能有效地提升网络传输效率。

GPB(Google Protocol Buffers)编码格式是一种独立于语言和平台的可扩展、序列化结构的数据格式,用于通信协议、数据存储等。它的主要优点是解析效率高,传递相同信息所占字节小。GPB 的编解码效率是 JSON 的 2～5 倍,编码后数据的大小是 JSON 的 1/2～1/3,既保证了 Telemetry 的数据吞吐性能,同时也节省了 CPU 和带宽资源。和一般的输出格式 XML、JSON 相比,GPB 是二进制格式,可读性比较差,所以 GPB 主要用于机器解析,以便更高效进行传输。

GPB 编码解析前:

```
{
  1:"NODE-A"
  2:"s4"
  3:"ifm:ifm/interfaces/interface"
  4:46
  5:1515727243419
6:1515727243514
  7{
   1[{
    1: 1515727243419
```

```
    2{
     5{
       1[{
     5:1
       16:2
     25:"Eth-Trunk1"
       }]
      }
     }
   }]
  }
  8:1515727243419
  9:10000
  10:"OK"
  11:"CE6850HI"
  12:0
}
```

GPB 编码解析后：

```
{
   "node_id_str":"NODE-A",
   "subscription_id_str":"s4",
   "sensor_path":"ifm:ifm/interfaces/interface",
   "collection_id":46,
   "collection_start_time":"2018/1/12 11:20:43.419",
   "msg_timestamp":"2018/1/12
11:20:43.514",
   "data_gpb":{
    "row":[{
     "timestamp":"2018/1/12 11:20:
43.419",
     "content":{
      "interfaces":{
       "interface":[{
         "ifAdminStatus":1,
         "ifIndex":2,
         "ifName":"Eth-Trunk1"
       }]
      }
}
    }]
},
```

```
    "collection_end_time":"2018/
     1/12 11:20:43.419",
      "current_period":10000,
      "except_desc":"OK",
      "product_name":"CE6850HI",
      "encoding":Encoding_GPB
}
```

3. 采集协议

gRPC 是一种高性能、开源和通用的 RPC 框架，面向移动应用和 HTTP2 设计，支持多语言，支持 SSL 加密通道。它本质上是提供了一个开放的编程框架，不同厂商都可以基于此框架，采用不同语言开发自己的服务器处理逻辑或客户端处理逻辑，从而缩短产品对接的开发周期。gRPC 协议栈分层如图 5-12 所示，各层的含义如表 5-3 所示。

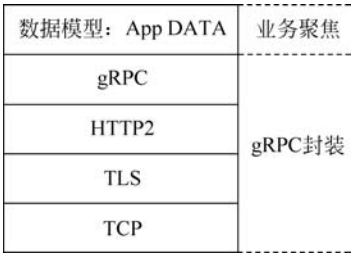

图 5-12　gRPC 协议栈分层示意图

表 5-3　gRPC 协议栈分层

层　　次	说　　明
TCP 层	底层通信协议，基于 TCP 连接
TLS 层	该层是可选的，基于 TLS 1.2 加密通道和双向证书认证等
HTTP2 层	gRPC 承载在 HTTP2 协议上，利用了 HTTP2 的双向流、流控、头部压缩、单连接上的多路复用请求等特性
gRPC 层	远程过程调用，定义了远程过程调用的协议交互格式
数据模型层	通信双方需要了解彼此的数据模型，才能正确交互

NETCONF(Network Configuration Protocol)是一种基于 XML 的网络管理协议，它提供了一种可编程的、对网络设备进行配置和管理的方法。NETCONF 协议采用基于 TCP 的 SSHv2 进行传送，以 RPC 的方式实现操作和控制。用户可以通过

NETCONF 协议设置参数、获取参数值、获取统计信息等。NETCONF 协议栈分层如图 5-13 所示，各层的含义如表 5-4 所示。

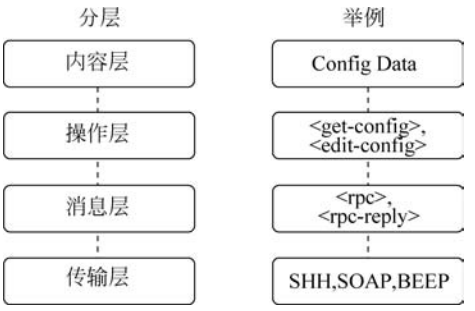

图 5-13　NETCONF 协议栈分层示意图

表 5-4　NETCONF 协议栈分层

层　　次	说　　明
内容层	内容层表示的是被管对象的集合，主要来自数据模型，采用 YANG 表示
操作层	四个层次中最核心的是操作层，主要包含查看配置、修改配置和会话加锁等操作，这些操作组成了 NETCONF 的基本能力
消息层	RPC 层主要作用是接收经过 NETCONF 传输层传递过来的 XML 格式的请求报文和回传响应报文，进行对应的解析和封装
传输层	传输层的主要作用是为 NETCONF 代理端和管理端的通信提供一个安全通道，给出 SSH、SOAP 和 BEEP 三种面向连接的通信传输协议，其中 SSH 是要求强制实现的协议

IPFIX(IP Flow Information Export)是由 IETF 公布的用于网络中流信息测量的标准协议，通过使用单一和一致的模型，简化了流输出架构，统一了流量监控标准。

随着 IPFIX 标准的广泛采用，网络管理员不用再担心如何支持多个流报告应用，每个应用都会拥有自己的流输出格式，IPFIX 让网络管理员可以使用一个符合该项标准的流报告应用程序。此外，IPFIX 的可扩展性使得网络管理员不必在传输流监测或报告需求发生变化时修改或升级设备配置。

4. 传输协议

与上述采集协议配套的传输协议主要有 TCP、UDP、HTTP 等标准协议，分别适用于不同的场景。例如管理面数据采集对传输可靠性要求较高，可以采用 TCP 传输；数据面对传输效率要求较高，可以采用 UDP 传输。

5.2.3　数据湖

数据湖汇聚全网各点、各层、各时期的数据，让海量的网络数据能够融合管理，融合分析。为解决网络中大规模的数据存储和访问难题，数据湖面临着多项技术挑战。一方面，多地域、多层次的网络数据容易形成孤岛效应，需要设置统一的存储资源池消除数据孤岛。另一方面，长周期的网络数据存储要求更高的存储利用率，支持弹性按需扩展，同时保证高可靠性。因此，数据湖技术在存储架构上需要满足如下特点。

（1）分布式架构：大数据存储采用分布式的架构，包括分布式管理集群、分布式哈希数据路由算法、分布式无状态集群和分布式智能 Cache 等，这种架构使得整个存储系统没有单点故障。

（2）高性能和高可靠性：大数据存储在所有磁盘中实现负载的均衡，数据打散存放，不会出现热点，采用高效的路由算法和分布式 Cache 技术以保证高性能及高可靠性要求。

（3）并行快速故障重建：数据分片在资源池内打散，硬盘故障后，可在全资源池范围内自动并行重建，重建效率高。

（4）易扩展和超大容量：大数据存储的分布式无状态集群可横向扩展，存储与计算分别按需平滑扩容，支持非烟囱式超大容量扩展。

数据湖的逻辑架构如图 5-14 所示，其部署时可采用分布式数据处理系统，提供企业级大数据存储、查询、分析的统一平台，实现海量规模数据信息的处理。通过对数据信息实时与非实时的分析挖掘，可发现全新价值点和企业商机。

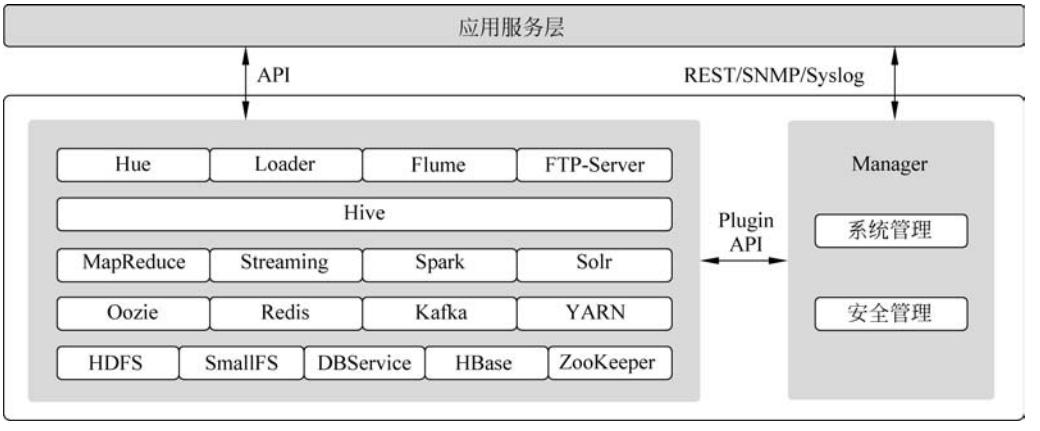

图 5-14　分布式数据湖架构图

分布式数据处理系统需要对开源组件进行封装和增强，对外提供稳定可靠的数据存储、查询和分析能力。不同组件提供的功能如下所述。

（1）Manager：作为管理者，为分布式数据处理系统提供高可靠、安全、容错、易用的集群管理能力，支持大规模集群的安装/升级/补丁、配置管理、监控管理、告警管理、用户管理、租户管理等。

（2）HDFS：Hadoop 分布式文件系统（Hadoop Distributed File System），提供高吞吐量的数据访问功能，适合大规模数据集的应用。

（3）ZooKeeper：提供分布式、高可用性的协调服务能力，帮助系统避免单点故障，从而建立可靠的应用程序。

（4）YARN：Hadoop 2.0 中的资源管理系统，它是一个通用的资源模块，可以为各类应用程序进行资源管理和调度。

（5）MapReduce：提供快速并行处理大量数据的能力，是一种分布式数据处理模式和执行环境。

（6）Spark：基于内存进行计算的分布式计算框架。

（7）Hive：建立在 Hadoop 基础上开源的数据仓库，提供类似 SQL 的 Hive QL 语言操作结构化数据存储服务和基本的数据分析服务。

（8）Kafka：一个分布式、分区、多副本的实时消息发布-订阅系统，提供可扩展、高吞吐、低延迟、高可靠的消息分发服务。

（9）DBService：一个具备高可靠性的传统关系型数据库，为 Hive、Hue、Spark 组件提供元数据存储服务。

5.2.4 大数据部署方案

在全光自动驾驶网络场景中，为了对网络数据进行正确、全面的感知和采集，获取网络物理状态的大数据，采用三级分层处理，构建从网元层、网络管控层到云端的完整数据链，如图 5-15 所示。

（1）网元层：部署遍布全网的光 Sensor，获取数字化的光网络物理层参数，对光功率、BER、SOP 等性能直接测量。这类数据的特点是实时性强，信息量大，精度要求高，要求处理速度达到毫秒级。光 Sensor 与单板间采用高速总线，网元内实现毫秒级采样，并对采集的数据进行清洗、生命周期管理，支撑本地高速智能决策。

（2）网络管控层：对网元的告警性能、拓扑与业务路径、光链路状态和网络参数等进行分析，采用高效实时的 Telemetry 数据采集协议，并借助高速 DCN 专用物理通

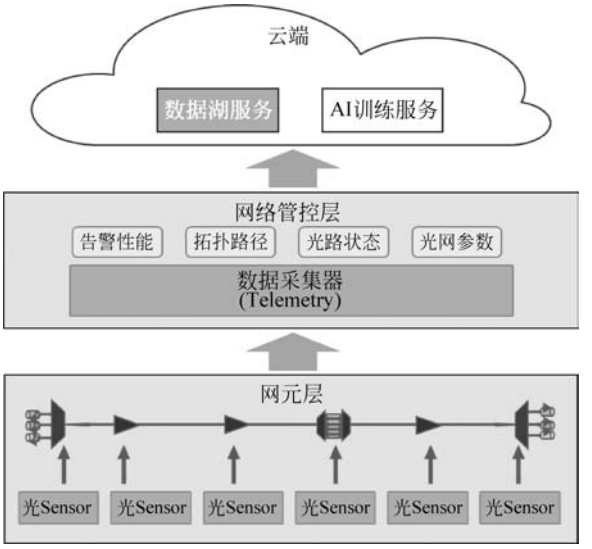

图 5-15　大数据技术部署示意图

道，实现从网元层到网络管控层的秒级数据采集，支撑集中化智能决策。

（3）云端：建立统一数据湖和统一数据标准，将海量部署的网络设备中积累的历史数据进行统一存储，作为模型训练的优质数据库。统一数据湖不仅能提供全量数据的完整生命周期管理，而且还能提供数据的分析、使能服务。

5.3　算力技术

算力是智能运维的基础能力，处理海量的网络数据离不开算力保障。AI 技术从最初的通用 CPU 计算开始起步，到基于 GPU 计算获得较大发展，当前已经发展到使用专用 AI 芯片提供高性能算力的阶段。通过专用 AI 芯片，为大数据采集、存储、分析、训练、上报等使用，提供十分强大的算力服务。全光网络智能化的算力，需要在云端和本地设备合理部署，通过算力架构的协同，实现智能计算的实时性、有效性和精确性。

5.3.1 设备侧算力技术

全光自动驾驶网络对系统算力的要求体现在如下两个方面。

(1) 相对于传统系统,海量待处理的原始检测数据以及愈加复杂的 AI 算法,都要求全光自动驾驶网络具有强大的计算能力。例如,神经网络 AI 算法本质上就是一系列复杂的数学计算过程。

(2) 全光自动驾驶网络需要应对低时延、敏捷响应的场景,业务处理要求达到毫秒甚至微秒级别(例如业务倒换时),必须具备足够算力资源才能满足系统高实时性的基本要求。

全光自动驾驶网络将依托集成 AI 专用计算芯片、AI 平台、分布式系统等软硬件,打造设备侧算力。

1. 专用 AI 芯片

AI 应用对算力的要求与 AI 算法种类以及算法复杂度有关。对于神经网络类 AI 应用算法,本质就是典型的数学计算问题,如果使用通用 CPU 核处理效率低,不但难以满足计算性能要求,而且会耗费边缘网络设备中宝贵的 CPU 资源,导致正常的业务受到影响。

专用 AI 芯片一般用来加速 AI 应用,特别是基于神经网络的深度学习应用的硬件,其部署方式可以是独立芯片封装,也可以是集成在其他器件内的计算单元,或者是独立的模块。基于不同的应用场景,业界已经推出的边缘计算 AI 芯片包括英伟达(Nvidia)推出的 Jetson Nano TX1/TX2、谷歌(Google)推出的 Edge TPU、华为推出的昇腾 310/910 系列等。

以英伟达面向嵌入式场景的 Jetson Nano SOC 为例,内部除集成了 4 核 Cortex-A57(1.4G 64 位)之外,为提升 AI 计算能力,还集成了专用于神经网络计算的 128-core NVIDIA Maxwell CUDA CORE。其 CUDA CORE FP16 算力可以达到 472GFOPS。

2. 分布式计算

分布式系统是将多台单机联网协同工作,在单机能力基本不变的情况下,通过增加可用的单机系统数量,将单机系统的计算能力和存储能力得到数量级的提升。

对于全光自动驾驶网络架构中的边缘计算节点,虽然从网络架构上看属于单个节

点，但是内部组成上并不都是单机系统，尤其是位于骨干网络边缘的节点设备，内部往往是由多个通过总线互联的插卡组成的复杂单元。可以将这些相互连接的插卡视为一个独立的小型分布式计算系统，统一管理系统内各个插卡上的计算资源，满足边缘计算节点对算力的具体要求。

如图 5-16 所示，整个机框上存在多个插槽，所有插槽的插卡通过以太总线相连接，多个机框再通过以太网线连接组成一个节点。

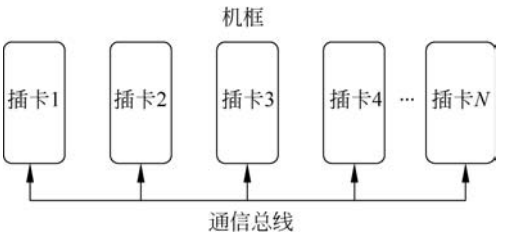

图 5-16　机框图

（1）算力线性扩展。通过分布式管理系统将各个插卡上的计算资源（CPU、AI 芯片、DSP……）统一管理、集中分配，不仅可以充分利用现有设备的剩余算力，还可以通过增加专用计算插卡的方式让设备侧算力得到成倍提升，满足智能光网对算力的要求。

（2）硬件特性和算法匹配。不同算法和场景的应用条件不同，导致算力需求迥异。神经网络算法计算量大，对数学计算能力提出严格要求；数据预处理算法需要频繁访问数据，更加关注内存以及 Cache 性能。训练场景下计算时间长、计算量大，对计算精度敏感（一般要求 FP32）；推理场景下要求的计算精度和计算量降低，但是实时性要求要高。面对差异化的算力需求，只有将算法、场景和硬件特性进行严格匹配，才能最大限度地发挥硬件优势，实现综合性能最佳。

（3）不同插卡上计算负载均衡。插卡上总的计算能力受限于硬件，对应一个固定值，但是满足固有特性之外的闲置计算资源仍处于动态变化过程中。通过实时监控和统计各个插卡上 CPU、内存、磁盘、通信带宽等硬件资源的实时变化数据，匹配 AI 算法的计算量以及插卡当前所能提供的空闲资源，可以将 AI 计算任务调度到对应的插卡上。

分布式系统计算任务调度既需要考虑算法特征/硬件特性的静态匹配，又需要支持资源实时变化和算法计算量的动态匹配，如图 5-17 所示。

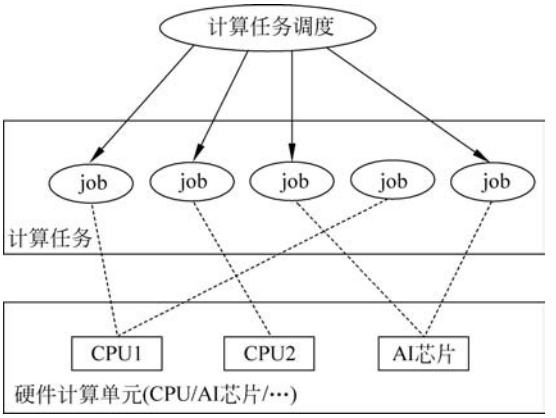

图 5-17　分布式系统计算任务调度示意图

5.3.2　云端算力技术

全光自动驾驶网络要求在云端和设备侧合理部署算力资源。设备侧以实时性要求高的数据分析和应用为主，需要部署高效计算并且低功耗的 AI 芯片及加速板卡。云端以高强度模型训练和在线推理为主，需要部署高计算密度的 AI 芯片。

云端模型训练涉及海量的训练数据、复杂的算法结构、迭代的学习过程等，推理优化也会涉及大批量数据处理或流式数据处理，因此需要采用一定规模的 AI 芯片集群来保障云端充裕的算力供给。另外，云端智能也可以通过软件即服务(Software as a Service，SaaS)方式提供 AI 应用，处理实时性要求不高的数据。

AI 芯片从技术架构上可以分为通用类芯片(CPU、GPU、FPGA)、基于 FPGA 的半定制化芯片、全定制化 ASIC 芯片等，如表 5-5 所示。

表 5-5　AI 芯片对比

芯片架构	算　力	应用场景	优　　劣	开发特点
CPU(中央处理器)	低	数据处理，如数据格式转换、数据标注	并行算力差，难以满足深度学习要求	优化软件提升硬件效率
GPU(图形处理器)	中	云端训练、边缘训练、云端推理、工业终端推理	并行算力好，价格昂贵，功耗高，面积大，推理侧应用受限	
FPGA(现场可编程门阵列)	中	更适合根据特定算法进行本地化的定制推理	架构灵活，可根据算法适配，但总体算力一般	优化硬件释放算法潜力
ASIC(专用集成电路)	高	全场景	针对人工智能专门设计，性能功耗可以达到最佳，但后续算法若有变化，无法适配	

通用计算发展的步伐正逐渐放缓，以 2008 年为转折点，芯片的一个重要衡量指标性价比从(Performance/Dollar)之前的每年平均递增 48%降低到了 10%以下，已难以满足市场快速增长的算力需求。算力的突破从通用 CPU 转向了专用芯片，GPU 最早满足了海量并行计算的基本诉求，其特点如下所述。

(1) 计算结构适合并行处理(Parallelism)。

(2) 计算模式构建齐整规律(Regularity)。

(3) 不需要过多地访问内存(Locality)。

(4) 可用更低精度(Precision)等效替代。

GPU 具备高并行结构，且拥有更多的逻辑运算单元(Arithmetic Logic Unit，ALU)，更适合对密集型数据进行并行处理，程序在 GPU 系统内的运行速度相较于单核 CPU 往往提升几十倍乃至上千倍。

然而 GPU 提供的算力资源更具通用性，并非 AI 专用算力。一方面，GPU 结构不是专门针对神经网络结构设计的，在 AI 计算时会浪费大量的无用面积(Dark Silicon)。另一方面，它尚不支持物理资源切区和真正的多租户，在提高云的利用率和实现真正的弹性方面没有太大灵活性。GPU 的驱动程序更像是一个黑匣子，最终用户缺乏主导权。因此，实现 AI 计算加速还需要更加专业化的芯片服务。TPU(Tensor Processing Unit)是一款用于云端的 AI 专用芯片，通过大规模脉动阵列结合大容量片上存储方式来高效加速深度神经网络中最为常见的卷积运算，架构相对简单。但 TPU 过于专用，云端芯片市场仍需要 GPU 作为其他场景用途的补充。

同时，不同的芯片架构各有擅长的计算模式与对性能功耗设计的权衡考虑，而具体任务的计算需求往往是混合的，因此面向全场景的异构计算问题也是当前 AI 算力技术研究的热点。

5.4　自动化协议技术

传统静态网络在业务发放、可靠性(光纤故障)和运维效率(依赖人工)上面临一系列的挑战。

(1) 传统网络运维效率低下，业务发放或调整时需要管理人员提前人工规划或优化路径，并通过网管逐个单站配置，开通时间长，且业务时延需经过大量人工计算和反

复测试，人力投入大，成本高。

(2) 传统网络可以对抗单点失效和一次断纤事件，但无法应对并发多重故障场景，难以满足高价值业务可靠性需求。

自动化协议技术可有效解决上述问题，推动传统静态网络向自动化网络组织方向演进。光网络自动化协议主要包括南北向协议、自动化控制协议和自动部署协议。引入自动化协议技术可减轻网络维护管理压力，自动建立端到端路径，缩短业务配置时间，多处断纤故障后可自动恢复，提升业务可靠性，统一大网管理，降低运维成本。

5.4.1 北向协议

网络管理系统通过开放北向接口以供上层运营支撑系统(Operations Support System，OSS)调用，使得 OSS 能够实现在自身的网管系统上对厂家设备进行管理和控制。

就像手机充电口从原始仅支持充电功能的圆形接口发展到后来的 Mini USB 接口、Micro USB 接口，再到现在标配的 Type-C 接口，网络北向接口技术同样也在不断演进，经历了从 SNMP、CORBA、XML、TEXT、交易语言 1(Transaction Language 1，TL1)到 REST 的不同发展阶段。同一个 Type-C 接口充电器可以给各种品牌的手机充电，是因为手机制造商均是按照同一套标准设计充电口。类似地，北向接口也需要满足一定的协议标准，才能与各种各样的 OSS 和不同厂家的设备完成对接。

可见，网络北向接口设计标准化是为了实现全网一体化运维的目标，运营商的综合网管平台(即上层 OSS)需要把各厂家光网络设备网管的信息收集起来统一呈现，所以要求不同厂家设备的网管接口遵循规范的命令格式。不同运营商对北向接口标准的要求有所区别，设备商网管系统(如 NCE)的北向接口需要满足与上层 OSS 对接要求。

图 5-18 展示了网络北向接口的演进历程。从图中可以看出，电信管理论坛(Telecommunication Management Forum，TMF)推出的多技术操作系统接口(Multi-Technology Operations System Interface，MTOSI)2.0 距今已超过 10 年，其更新进展缓慢，同时无法兼容新的标准。2008 年后因特网工程任务组(Internet Engineering Task Force，IETF)在本领域发展迅猛，Netconf/Yang 的标准化契合 SDN 以及 OpenDaylight 产品化迅速得到推广和认可，成为新一代接口的事实标准。

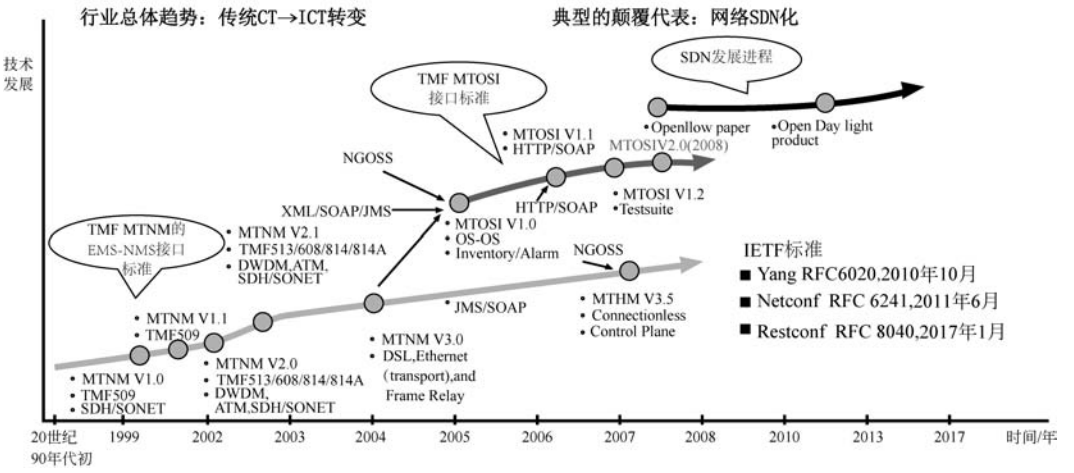

图 5-18　北向接口演进历程示意图

1. CORBA 接口

CORBA 是一个分布式的面向对象应用架构规范，是一种异构平台下的与语言无关的对象互操作模型。换句话说，CORBA 设计是独立于平台和语言的，因此它能够在任何平台上运行，定位于网络的任何地方，使用任何有接口定义语言（Interface Definition Language，IDL）映射的语言。它的核心是一套标准的语言、接口和协议，以支持异构分布应用程序间的互操作性及独立于平台和编程语言的对象重用。

如图 5-19 所示，CORBA 北向接口主要包括三部分：标准的 CORBA 名字服务、通知服务和 CORBA 代理（Agent），具体如下所述。

（1）CORBA 名字服务：为 OSS 正确地访问 NMS CORBA 接口提供了唯一的入口。在部署时，需保证被管理 NMS 的名字在 NMS 管理域内的唯一性。

（2）CORBA 通知服务：负责添加/删除事件监听，并接收告警/性能/通知。在 NMS 中配置数据发生变更时，能及时将详细的变更信息通知 OSS，保证 OSS 和 NMS 数据的一致性。

（3）CORBA 代理：负责将 NMS 的内部数据转化为符合国际标准的 CORBA 数据，以及将 OSS 请求信息转化为 NMS 系统内部交互信息，保证运营商能够按照标准进行对接，降低对接成本。

CORBA 接口功能如表 5-6 所示。

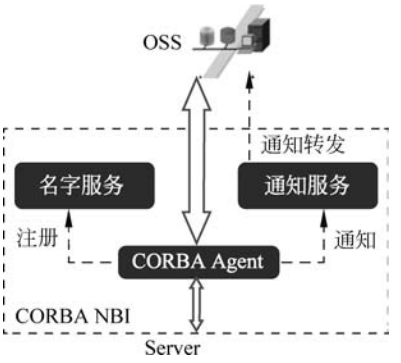

图 5-19　CORBA 关键组件

表 5-6　CORBA 接口功能

介　绍　项	说　　明
定位和使用场景	CORBA 接口在传送领域支持单站和端到端方式业务发放以及告警性能管理，IP 和接入领域仅支持基础物理存量和告警功能
接口协议	遵循的业务标准：TMF MTNM V3.5，TMF513/608/814 遵循的技术标准：OMG CORBA 2.6，IIOP 1.1 & IIOP 1.2，Naming Service 1.1，Notification Service 1.0
兼容、演进策略	已经集成的局点可继续演进，新集成局点不推荐

2. XML 接口

北向 XML 接口是遵循电信管理论坛(MTOSI)的开放接口。通过北向 XML 接口可以为各级网络管理系统提供统一的互联通道，有利于程序之间的交互。采用北向 XML 接口可以很好地适应网络管理综合化、跨域化的发展趋势。

XML 接口功能如表 5-7 所示。

表 5-7　XML 接口功能

介　绍　项	说　　明
定位和使用场景	XML 接口支持的功能和设备领域比较全面，多用于传送领域和 IP 领域的业务发放、告警上报、当前性能查询等运维场景
接口协议	遵循的业务标准：TMF MTOSI V2.0，TMF518/612/864 遵循的技术标准：SOAP 1.1，WSDL 1.1，JMS v1.1，HTTP(S)1.1
兼容、演进策略	已经集成的局点可继续演进，新集成局点不推荐

3. TEXT 接口

性能文本(TEXT)接口是典型的北向接口技术之一,主要用于生成北向性能文本文件,与上层 OSS 进行性能管理集成。TEXT 接口功能如表 5-8 所示。

表 5-8　TEXT 接口功能

介　绍　项	说　　明
定位和使用场景	TEXT 接口支持传送、IP、接入领域的历史性能周期导出功能,由于历史性能数据量大,文本集成方式效率要远高于 CORBA、XML 接口的按需查询响应方式,所以历史性能推荐使用 TEXT 接口导出
接口协议	—
兼容、演进策略	新 NMS 性能文本北向接口兼容老 NMS 版本已经提供的性能文本接口功能,已经集成的局点可继续演进,新集成局点的历史性能集成推荐使用性能文本(FTP 性能)方式集成

4. REST 接口

当前,北向协议接口主流技术是 RESTful 接口。RESTful 可视为一种软件架构的设计风格,只是提供了一组设计原则和约束条件,而非严格意义上的标准。它主要用于客户端和服务器交互类的软件。基于该风格设计的软件,更为简洁和具有层次性,更易于实现缓存等机制。

北向 REST 接口使用的是基于微服务架构和 RESTful 架构风格的技术。微服务架构是一项在云中部署应用和服务的新技术,通过轻量级的 Web 服务对外提供能力,其特点是通过 JavaScript 对象表示法(JavaScript Object Notation,JSON)和另一种标记语言(Yet Another Multicolumn Layout,YAML)定义服务和数据结构,使用安全超文本传输协议(Hypertext Transfer Protocol Secure,HTTPS)、服务器推送事件(Server-Sent Event,SSE)、WebSocket 等协议传输数据,并通过 RESTful 风格来管理网管资源。

1) REST 接口发展背景

2000 年,加州大学欧文分校的罗伊·托马斯·菲尔丁在他的博士论文 *Architectural Styles and the Design of Network-based Software Architectures* 中首次描述了 REST 架构风格和设计思想。菲尔丁博士还是超文本传输协议(Hypertext Transfer Protocol,HTTP)和统一资源标识符(Uniform Resource Identifier,URI)等 Web 架构标准,以及 Apache HTTP 服务器的主要设计者。

REST 是为了使 Web 可以高效运转而创建的一种架构模型，是 Web 协议标准的指导框架。符合 REST 原则的 HTTP 方法称为 RESTful API。

2）REST 接口特点

REST 接口有如下特点。

（1）以资源为基础，每个资源都可以通过 URI 访问。

（2）对资源的操作包括查询、创建、修改和删除，对应 HTTP 协议的 GET/POST/PUT/DELETE 方法。

（3）使用 XML/JSON/YAML 等作为传输报文格式。

REST 接口功能如表 5-9 所示。

表 5-9　REST 接口功能

介　绍　项	说　　明
定位和使用场景	REST 接口功能在逐步完善中，当前支持传送、IP、接入领域的组合业务的发放、资源管理、故障管理等功能
接口协议	参考的业务标准：IETF 遵循的技术标准：HTTP(S)1.1，json Service 1.0，YANG 1.0
兼容、演进策略	NMS REST 风格接口为新构筑的轻量级北向接口，新局点业务发放、故障管理、存量同步集成场景推荐使用

RESTCONF 是一种 RESTful 协议，提供 HTTP 上的编程接口，用于访问 YANG 定义的数据，使用的模型工具是 YANG 建模语言。在全光自动驾驶网络架构中，RESTCONF 作为北向协议接口实现 TSDN 控制器跟上层运营支撑系统的通信。

全光自动驾驶网络的北向接口基于设备、网络和业务三层模型，提供多种原子服务和场景驱动可编程平台，支持定制开发场景化的工作流，实现全自动或半自动化的意图设计、执行、优化流程，可满足对网络维护工作的简化需求。一般以 ACTN 接口作为标准，通过 RESTCONF/YANG 进行定义，并兼容既有的 RESTful 接口，能够快速地与 BSS/OSS 等上层应用系统集成，支持定制开发各类 App，加快业务创新和实现电商化运营。

RESTCONF 接口协议采用 HTTPS v1.1(RFC 2616)传输协议，端口号为 26335。

REST 从资源的角度来观察整个网络，分布在各处的资源由 URI(Uniform Resource Identifier)确定，而客户端的应用通过 URL(Uniform Resource Locator)来获取资源，如表 5-10 所示。

资源的 RESTful URL 采用以下模板：

```
https://ip:port/{prefix}/{service-name}/{version}/{rest-path}
```

表 5-10　RESTful URL 说明

组　　成	含　　义
ip：port	服务接口入口
prefix	前缀
service-name	服务名称
version	接口版本号，版本号使用“v＋正整数”，从 v1 开始，接口发生不兼容变更时会更新版本号
rest-path	服务内资源路径

RESTCONF 在采用 JSON 编码格式时要求如下。

(1) 请求/响应报文使用 JSON 报文(RFC4627)。

(2) 媒体类型表示为 application/json。

(3) 所有 API 均使用 UTF-8 编码。

(4) 因通用 JSON 解析算法不保序，故报文中同一层级的字段前后顺序不保证不变。

RESTCONF 请求方法符合 REST 风格，对资源进行的操作必须符合 HTTPS 规范定义，如表 5-11 所示。

表 5-11　RESTCONF 请求方法和资源操作

请 求 方 法	资 源 操 作
POST	请求服务器新增资源或执行特殊操作
GET	请求服务器返回指定资源
PUT	请求服务器全量更新指定资源
DELETE	请求服务器删除指定资源
PATCH	请求服务器部分更新指定资源

5.4.2　南向协议

南向协议是控制器与设备之间的通信协议，完成控制器对设备控制指令下发和网络设备资源与状态实时上报。南向协议主要包括 NETCONF、OSPF、PCEP、OPENFLOW 等类型的协议接口。

1. NETCONF 协议

NETCONF 是为弥补简单网络管理协议(SNMP)和 Telnet 协议在网络配置方面的功能不足所设计的一种用于网络数据设备配置管理的协议。

NETCONF 协议提供了安装、操作和删除网络设备配置的机制,其采用了基于数据编码的可扩展标记语言配置数据以及协议信息,在自动化网络配置系统中 NETCONF 起着关键性的作用。

2. PCEP 协议

PCEP 协议用于实现对 TSDN 网元的集中控制。NMS 作为路径计算设备(Path Computation Equipment,PCE),TSDN 网元设备作为路径计算客户(Path Computation Client,PCC),两者之间采用 PCEP 协议通信,获取设备资源信息,提供集中算路服务,并维护链路状态。

3. gRPC 协议

gRPC 协议(google Remote Procedure Call Protocol)是谷歌发布的一个基于 HTTP2 协议承载的高性能、通用化 RPC 开源软件框架。通信双方都基于该框架进行二次开发,从而能够聚焦业务本身,而无须关注由 gRPC 软件框架实现的底层通信。

gRPC 支持 GPB(Google Protocol Buffer)编码格式。GPB 是一种与语言无关、与平台无关、扩展性好的序列化结构数据格式,主要用于通信协议、数据存储等,属于二进制编码,性能好、效率高。GPB 通过".proto"文件描述编码使用的字典,即数据结构描述。用户可以利用 Protoc 等工具软件(如 protoc-3.0.2-windows-x86_64.exe 文件)根据".proto"文件自动生成代码(如 Java 代码),然后用户基于自动生成的代码进行二次开发,从而实现与设备的对接。

gRPC 协议栈分层如表 5-3 所示。

4. SNMP 协议

SNMP 可以应用在上下层网管之间,也可以应用在设备与网管之间,如图 5-20 所示。SNMP 北向接口遵循 SNMP v1/v2c/v3 标准,向上层 OSS 提供统一的告警管理功能。

SNMP 接口功能如表 5-12 所示。

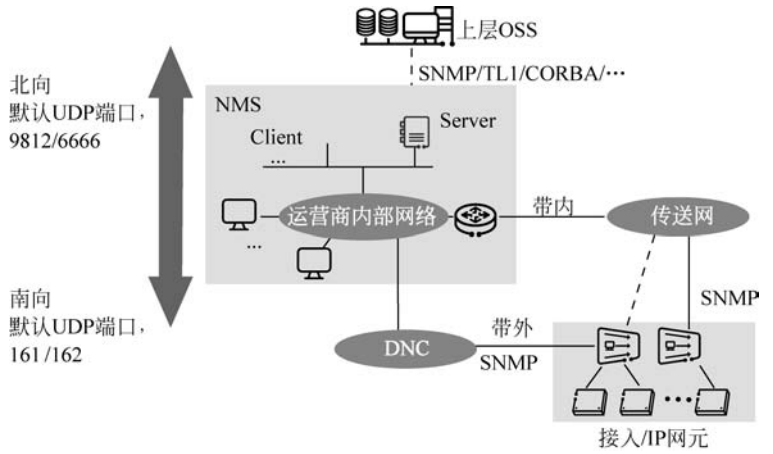

图 5-20　SNMP 接口的应用场景

表 5-12　SNMP 接口功能

介　绍　项	说　　明
定位和使用场景	SNMP 接口支持传送、IP、接入领域的告警上报和查询功能，但由于协议简单、功能单一，只有告警，无存量、发放等功能，且 SNMP 告警资源标识与 CORBA、XML 接口资源标识格式不一致，所以通常适用于仅要求集成告警功能的局点
接口协议	遵循的标准：SNMP v1/v2c/v3
兼容、演进策略	已经集成的局点可继续演进，新集成局点不推荐

5.4.3　自动化控制协议

目前，ASON 控制平面的主流技术是通用多协议标记交换（Generalized Multi-Protocal Label Switching，GMPLS）。作为在多协议标记交换（Multi-Protocal Label Switching，MPLS）基础上发展起来的一类技术，与 MPLS 侧重描述数据包的转发机制不同，GMPLS 更多关注连接管理的功能，并因此做了很多便于连接管理能力的扩展。采用 GMPLS 技术，可以对光纤/端口交换、波长交换、时隙交换、二层交换、包交换等不同类型的交换技术进行统一控制和管理。

控制平面在 ASON 中的主要任务是控制传送平面功能，进行连接过程的动态建立、拆除，以及对连接的维护和恢复。其功能可分为连接管理、资源管理、路由管理三大部分，分别由信令协议、链路管理协议、路由协议来完成。

引入控制平面的目的就是要使网络“智能”起来，这就要求能够自动发现邻居/资

源、自动计算路径、自动建立和管理连接(业务)。

1. 邻居发现

只有充分了解网络的节点、链路、带宽等信息,控制平面才能自动分配资源。实现网络的智能化,发现邻居及其相连的链路资源是基础。GMPLS 通过扩展 LMP 协议来进行自动邻居和链路发现,获得网络的资源信息,如图 5-21 所示。

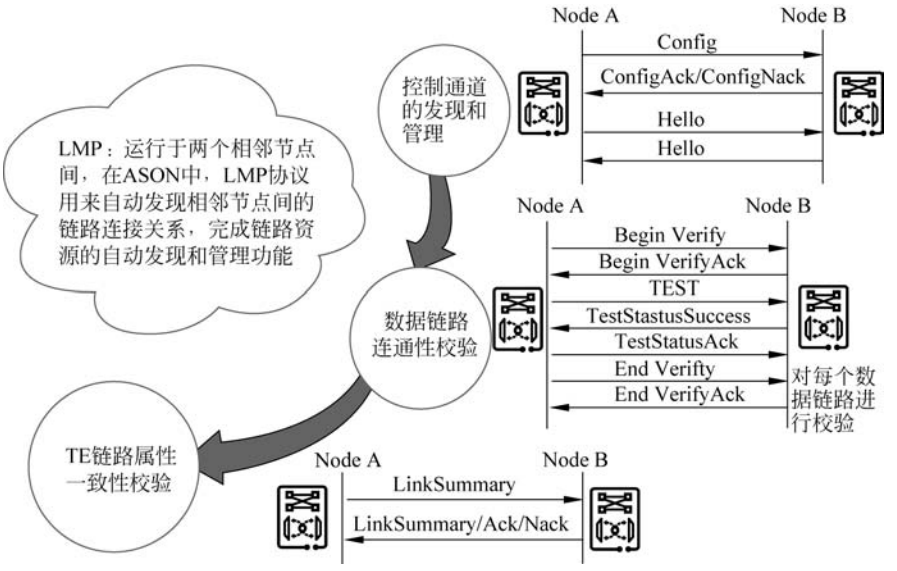

图 5-21　邻居发现过程

2. 拓扑发现

只知道邻居信息,还无法计算出端到端的路径,需要将邻居信息告知网络中的其他节点,通过相互通告,使得每个节点都知道全网的链路、资源信息,即拓扑信息。OSPF 协议和 IS-IS 协议是经典的域内路由协议。GMPLS 通过扩展 OSPF-TE、ISIS-TE 协议来进行资源扩散,以便获得全网的资源信息,生成拓扑,进行路径计算。目前,多数厂家使用 OSPF-TE 协议,如图 5-22 所示。

3. 业务创建

当从用户(网管/客户网络)接收到创建业务的请求后,首节点根据拓扑计算出端到端的路径,然后通过信令协议发起业务的建立。GMPLS 通过扩展 RSVP-TE、CR-

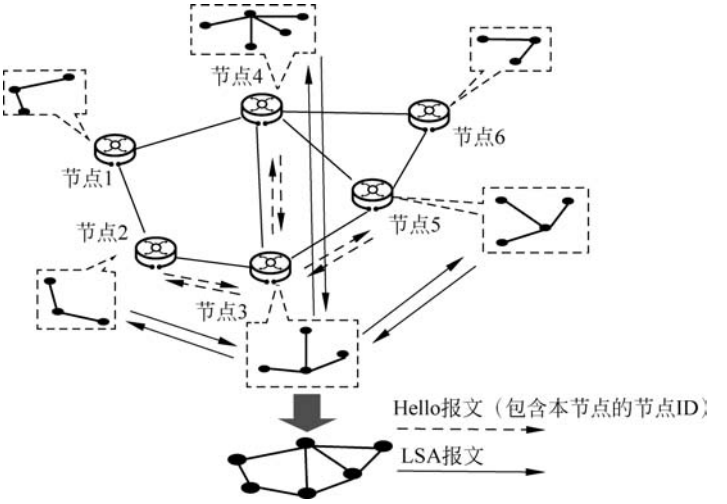

图 5-22　拓扑发现过程

LDP、PNNI 协议来实现业务的创建和管理，在实际应用中，多数厂家使用 RSVP-TE 协议，如图 5-23 所示。

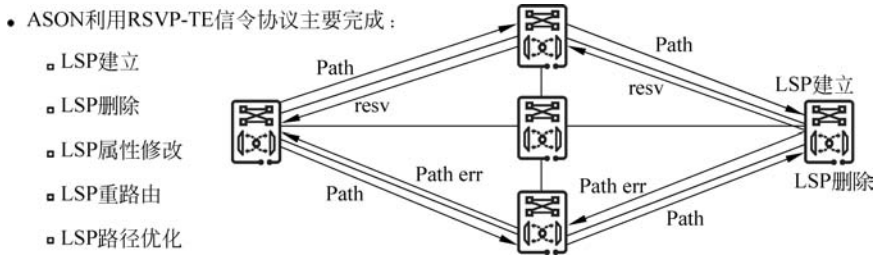

图 5-23　业务发现和路由协议

4. 业务故障后自动恢复（重路由）

在业务故障（如光纤中断）后，发现故障的节点将受影响的所有业务通过信令协议消息通告给首节点，首节点重新为每条业务计算路径，并通过信令协议按新的路径建立好业务连接，将业务切换到新的路径上进行传送，恢复业务，如图 5-24 所示。

5. 故障消失后业务自动返回

在业务原路径上的故障消失（如光纤修复）后，发现故障消失的节点将受影响的所有业务通过信令协议消息通告给首节点，首节点将这些业务切换到它们的原始路径

上，并删除它们的恢复路径，从而将业务返回到原始路径上进行传送，如图 5-25 所示。

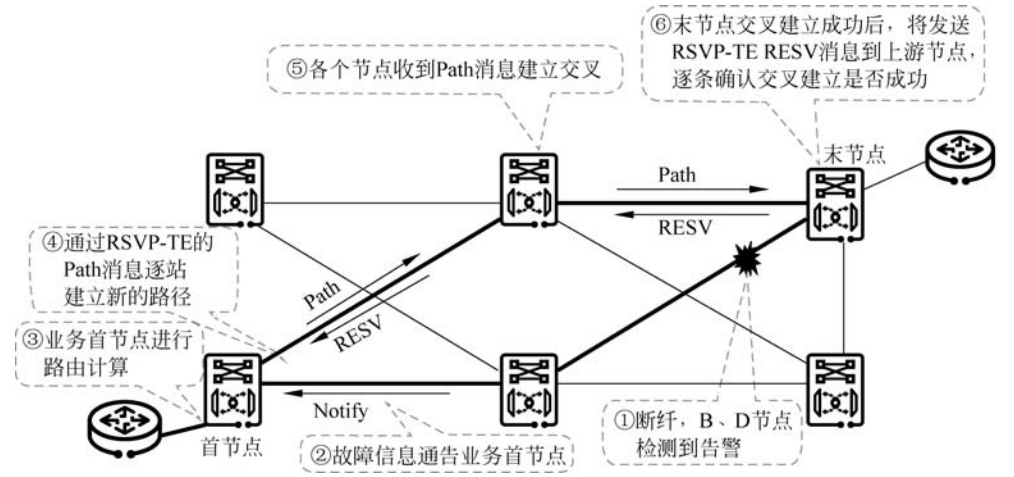

图 5-24　业务故障后自动恢复

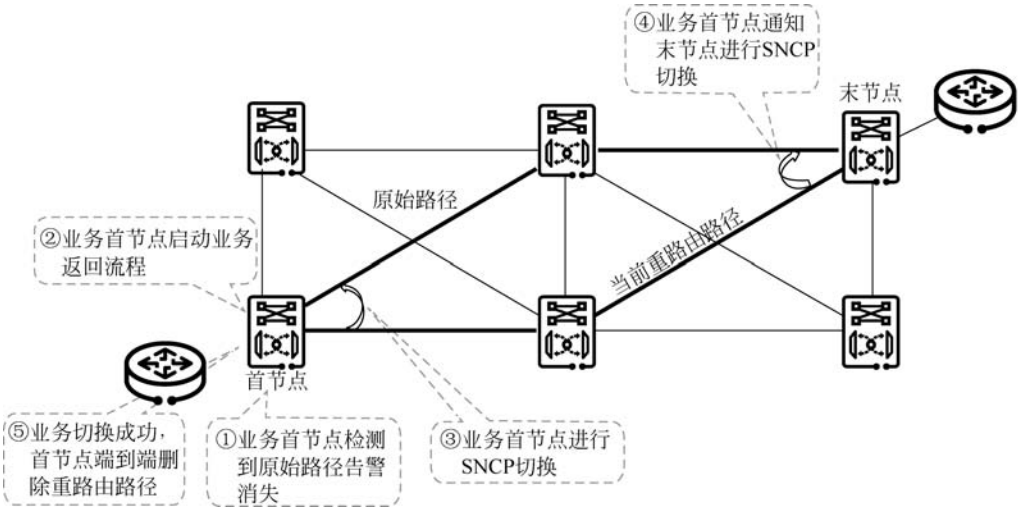

图 5-25　故障消失后业务自动返回

5.4.4　OLT 自动部署协议

在光接入网建设中，由于初始安装的 OLT 设备都是统一发货的，无法识别到该设备具体安装到什么运营商的什么站点上，所以新安装部署的 OLT 等设备也无法预先配置好管理的 IP 地址，按照传统的建设方式，必须要软件调测工程师携带便携机等到

OLT 站点本地，配置该 OLT 设备的管理 IP 地址和管理通道等，才可以支持在远程对该 OLT 设备的管理。

而智简光网络技术对传统的部署方案做了改进，实现了不需要软件调测工程师参与的 OLT 自动部署新能力。这也要求 OLT 设备能够支持相关的 IETF RFC 自动化部署协议。IETF RFC 8572 Secure Zero Touch Provisioning(SZTP)是一种在网络设备以出厂默认状态引导时安全配置网络设备的技术。该方案的变体使得它既可用于公有网络，也可用于私有网络。配置步骤能够更新启动映像、提交初始配置和执行任意脚本以满足辅助需求。更新的设备随后能够与其他系统建立安全连接。例如，设备可以与特定的网络管理系统建立 NETCONF(RFC 6241)和/或 RESTCONF(RFC 8040)连接。

IETF RFC8071 定义了 NETCONF 主动注册机制 Call Home 解决 OLT 设备初始安装之后的敏捷部署问题。Call Home 定义了 NETCONF Server 和 NETCONF Client 之间的信息交互功能，即 NETCONF Server(OLT 网元设备)主动发起 TCP 连接到 NETCONF Client，NETCONF Client 依据该连接建立 SSH、TLS 等安全通信通道，最终完成 NETCONF 通信。

图 5-26 中 OLT 的自动部署方案主要包括如下流程。

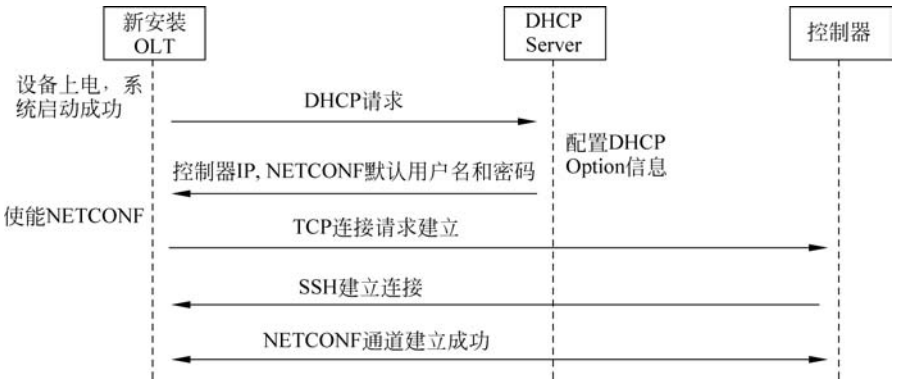

图 5-26　OLT 自动部署流程示意图

(1) 控制器预配置 OLT 自动部署策略，包含合法设备名单等。

(2) OLT 设备初始上电，系统启动成功，判断是否为新安装设备。

(3) 确认是否新安装设备，发起 DHCP 请求，请求控制器 IP 地址、NETCONF 默认用户名和密码。

(4) OLT 从 DHCP Server 处获取到控制器 IP，NETCONF 默认用户名和密码。

(5) OLT 自动使能 NETCONF 能力。

(6) OLT 基于 Call Home 标准发起 TCP 连接请求。

(7) TCP 连接建立成功之后,控制器使用 TCP 连接,向 OLT 设备建立 SSH 会话。

(8) 控制器使用 SSH 会话,向设备发送 NETCONF 握手报文,进行 NETCONF 版本协商和能力集的交换,完成 NETCONF 通道的建立。

(9) NETCONF 通道建立后,控制器可直接下发 OLT 预配置和正常管理 OLT 设备。