在 Nordic 的蓝牙协议栈实现中，连接参数的更新是在主机层（Host Layer）的通用访问配置文件（GAP）层发起的。以下是详细的技术解析：

1. 协议栈分层与核心流程

蓝牙协议栈分为控制器（Controller）和主机（Host）两部分。连接参数的管理涉及多个层次：

链路层（Link Layer）：负责物理连接的建立、数据传输时序和射频控制，直接处理连接间隔、从机延迟等参数。

L2CAP 层：通过信令通道（Signaling Channel）传输连接参数更新请求（如Connection Parameter Update Request命令），但该请求的发起方是上层协议。

GAP 层：作为主机层的核心，负责管理设备角色（如主机 / 从机）、连接状态切换和连接参数配置，提供直接控制连接参数的 API 接口。

2. Nordic 的具体实现

在 Nordic 的 SoftDevice 协议栈中，连接参数更新的发起逻辑集中在 GAP 层：

API 触发：开发者通过调用sd\_ble\_gap\_conn\_param\_update函数（属于 GAP 层接口）主动发起参数更新请求。例如，从机设备可通过ble\_conn\_params\_init函数配置参数并自动触发更新流程。

事件驱动：当设备接收到连接参数更新请求（如来自主机的参数调整），GAP 层会通过事件回调（如BLE\_GAP\_EVT\_CONN\_PARAM\_UPDATE）通知应用层，并根据配置决定是否接受或拒绝。

参数协商：GAP 层负责协调主机与从机之间的参数协商。例如，从机通过 L2CAP 信令通道发送请求后，主机在 GAP 层处理该请求并返回响应。

3. 关键函数与配置

sd\_ble\_gap\_conn\_param\_update：直接调用该函数可向对端设备发送连接参数更新请求，参数包括最小 / 最大连接间隔、从机延迟、超时时间等。

ble\_conn\_params\_init：在从机端初始化连接参数管理模块，配置首次更新延迟、重试间隔等策略，内部会自动调用sd\_ble\_gap\_conn\_param\_update。

GAP 配置参数：通过 Kconfig 或 SDK 配置文件（如prj.conf）可预定义连接参数的默认值，例如：

c

config BT\_GAP\_AUTO\_UPDATE\_CONN\_PARAMS

bool "Automatically update connection parameters after connection"

default y

该配置使从机在连接建立后自动发起参数更新。

4. 传输路径与协议交互

GAP 层发起请求：应用层调用 GAP API 后，参数更新请求通过 L2CAP 信令通道（CID=0x0005）传输至对端设备的 GAP 层。

链路层执行：对端设备的链路层根据接收到的参数调整物理连接的时序。例如，若主机接受从机的参数请求，链路层会更新连接间隔和从机延迟。

事件反馈：更新结果通过 GAP 层事件（如BLE\_GAP\_EVT\_CONN\_PARAM\_UPDATE）通知应用层，开发者可在此处理成功或失败逻辑。

5. 实际应用场景

从机主动优化功耗：从机设备（如传感器）在连接建立后，通过 GAP 层 API 请求增大连接间隔以降低功耗，同时保持数据传输的实时性。

主机动态调整性能：主机（如手机）可根据业务需求（如音频流传输）通过 GAP 层触发参数更新，要求从机降低从机延迟以减少传输延迟。

多设备并发管理：在多连接场景下，GAP 层协调不同连接的参数配置，避免链路层资源冲突。

6. 协议规范与兼容性

蓝牙核心规范：连接参数更新的标准流程要求请求由主机发起，但 Nordic 的实现允许从机通过 GAP 层 API 主动发起，这属于协议扩展功能。

跨平台兼容性：由于不同设备的 GAP 层实现可能存在差异，建议在参数协商时设置合理的容错机制（如重试次数、超时处理）。

总结

Nordic 设备通过GAP 层提供的 API 接口发起连接参数更新请求，该请求通过 L2CAP 层传输至对端设备的链路层执行。这一设计将参数管理的灵活性交给开发者，同时确保与蓝牙协议规范的兼容性。开发者可通过 GAP 层的事件回调和配置参数，精细控制连接参数的协商过程，以满足不同应用场景下的功耗、延迟和稳定性需求。