# 1.更改日志和作者

1.0 版本于 2022 年 9 月 30 日发布

# 2. 目标和学习目标

物联网（Internet of Things，简称 IoT）是指全球数十亿台物理设备，它们现在连接到互联网，所有这些设备都在收集和共享数据，以提高人们的生活质量。例如，现在非常流行的智能手表可以收集佩戴者的数据，并与中央服务器共享数据，以监测佩戴者的健康状况。在本次作业中，您将有机会根据物联网的理念实施您自己的数据收集和共享网络版本。您的应用程序基于客户端-服务器架构，该架构由一台服务器和多个客户端（即边缘 IoT 设备）以及对等网络进行并发通信。您的应用程序将支持通常从现有边缘网络中找到的一系列功能，包括身份验证（边缘设备和中央服务器之间）、边缘设备侧的数据生成以及边缘设备和中央服务器之间的数据共享，以及在一个边缘设备和另一个边缘设备之间。您将为基于 TCP 和 UDP 的边缘网络设计自定义通信协议。

## 2.1 学习目标

完成此任务后，您将获得以下技能方面的专业知识：

1. 详细了解客户端-服务器和客户端-客户端（点对点）交互如何工作。

2. 擅长socket编程。

3. 对设计和实现通信协议有所见解。

# 3. 作业规范

作业值 20 分。该规范分为两部分。第 3.2 – 3.3 节介绍了第一部分，涉及边缘设备（即客户端）和中央服务器之间的基本交互，包括边缘设备通过 TCP 与中央服务器通信的功能。第二部分，在第 3.4 节中介绍，要求您实现附加功能，使两个边缘设备可以通过 UDP 以点对点方式直接相互交换视频文件。CSE 学生应实现这两种功能。作业包括2个主要模块，服务器程序和客户端程序。服务器程序将首先运行，然后是客户端程序的多个实例（每个实例支持一个客户端）。它们将从相同和/或不同主机上的终端运行。

## 3.1分配规范

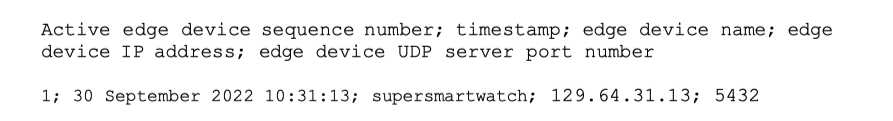
在此编程作业中，您将实现边缘网络的客户端（请注意，我们将交替使用客户端和边缘设备）和服务器程序，在许多方面类似于现有的边缘网络。您的应用程序将支持一系列操作，包括验证边缘设备并让边缘设备加入网络，在边缘设备生成数据，将数据从边缘设备上传到中央服务器，与中央服务器交互以获取计算服务，从中央服务器删除数据，读取活动边缘设备的信息，并将数据文件从一个边缘设备交换到另一台边缘设备（仅限 CSE 学生）。您将为边缘网络设计和实现一个通信协议来实现这些功能。服务器将侦听指定为命令行参数的端口，并等待客户端连接。客户端程序将启动与服务器的 TCP 连接。建立连接后，客户端将启动身份验证过程。客户端程序将通过命令行界面与用户交互。成功验证后，客户端将启动可用命令之一。所有命令都需要客户端和服务器或两个客户端（仅限 CSE 学生）之间的简单请求-响应交互。客户端可能会执行一系列命令（一个接一个）并最终退出。客户端和服务器都必须在命令提示符处打印有意义的消息，以捕获发生的特定交互。您可以自由选择显示的精确文本。客户端-服务器交互的示例在第 8 节中给出。

## 3.2 身份验证

当客户端请求连接到服务器时，例如加入边缘网络，服务器应提示客户端输入边缘设备名称和密码，并对边缘设备进行身份验证。有效的边缘设备名称和密码组合将存储在名为 credentials.txt 的文件中，该文件与服务器程序位于同一目录中。分配页面上提供了一个示例 credentials.txt 文件。边缘设备名称和密码区分大小写，您可以假设边缘设备名称对于此分配的上下文是唯一的。我们可能会使用不同的文件进行测试，因此请勿在您的程序中硬编码此信息。您可以假设每个边缘设备名称和密码将位于单独的一行，并且两者之间会有一个空格。设备名称和密码不包含任何空格。如果凭据正确，则认为边缘设备已成功验证并加入边缘网络，并显示欢迎消息。

输入无效凭据时，会提示客户端重试。在连续几次尝试失败后，此边缘设备被阻塞 10 秒（数字是提供给服务器的整数命令行参数，数字的有效值应在 1 到 5 之间）并且在这 10 秒内无法加入网络-秒持续时间（即使来自另一个 IP 地址）。如果向服务器提供了无效的数字值（例如，浮点值 0 或 6），则服务器会打印出一条消息，例如“允许的连续失败尝试次数无效：数字。参数号的有效值是 1 到 5”之间的整数。

对于 CSE 学生：边缘设备成功加入网络后，边缘设备（即客户端）接下来应向服务器发送它将侦听 P2P 连接的 UDP 端口号。服务器应在活动边缘设备日志文件 (edge-device-log.txt) 中记录边缘设备加入的时间戳、边缘设备名称、IP 地址和 UDP 端口号：



为简单起见，边缘设备将在任何给定时间加入网络一次，例如，不允许多个同时加入，我们不会测试这种情况。

EDG：边缘数据生成

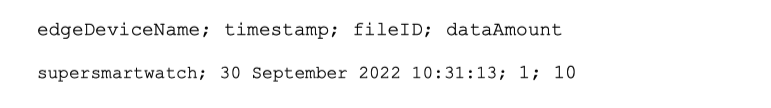
EDG fileID dataAmount

fileID 和 dataAmount 应作为参数包含在内，fileID 是一个整数，用于唯一标识将用于存储生成数据的文件，文件名应为边缘设备名称-fileID 和文件类型应为 txt（例如 supersmartwatch-1.txt）。 dataAmount 参数用于指示要生成的数据样本的数量。可以随机生成数据样本，对数据的生成没有严格要求，例如，如果dataAmount指定为10，那么可以生成任意10个整数（例如，从1到10，从20到30，或者任意一个）其他 10 个整数）并将它们存储到数据文件中。另外，在将数据样本存入文件时，应遵循“一行一号”的规则，提供了一个示例（supersmartwatch-1.txt）。请注意，如果该文件已存在于后续 EDG 命令调用中，则应直接用新生成的数据样本覆盖现有数据样本。如果 EDG 命令中缺少 fileID 或 dataAmount 参数，则应提示适当的错误消息，例如，“EDG 命令需要 fileID 和 dataAmount 作为参数。”，如果提供的 fileID 和 dataAmount 参数不是整数，则应提示适当的错误信息，例如“fileID或dataAmount不是整数，您需要将参数指定为整数”。

边缘设备成功生成数据样本并将其存储到文件中后，应提示适当的消息（例如，“数据生成完成”）以指示该命令已被边缘设备成功处理。

UED：上传边缘数据

UED fileID 边缘设备将要上传的特定文件的文件ID 作为参数包含在内。边缘设备（即客户端）收到此命令后，应从相应文件中读取数据样本，并使用 TCP 将数据样本传输到中央服务器。客户端需要检查是否提供了fileID参数，如果缺少fileID，您应该提示例如“需要fileID来上传数据”的消息。客户端还需要检查对应的文件是否存在，如果边缘设备端不存在该文件，则提示“要上传的文件不存在”。中心服务器成功接收文件后，服​​务器应向边缘设备发送消息，通知服务器已成功接收文件，客户端也应提示适当的消息，表明文件上传成功。此外，中央服务器应维护一个名为uploadlog.txt 的上传日志文件。如果一切正常，服务器应附加以下格式的上传日志消息：



SCS：服务器计算服务

SCS fileID 计算操作

该命令旨在请求强大的中央服务器进行各种计算，因为实际上边缘设备通常具有非常有限的计算资源。 fileID 和 computeOperation 作为两个参数包含在内。 fileID 用于指示用于计算目的的相应数据文件。如果未提供 fileID 或 fileID 不是整数，客户端应提示适当的错误消息，例如“fileID is missing or fileID should be an integer”。服务器还需要检查相应的文件是否存在，如果文件不存在，则服务器应向客户端响应消息，通知客户端该文件不存在，客户端应提示正确的消息指示该文件不存在。文件在服务器端不存在。为简单起见，我们为此分配定义了总共四个计算操作：SUM、AVERAGE、MAX、MIN。 SUM——计算对应文件中数据样本的总和，AVERAGE——获取对应文件中数据样本的平均值，MAX——获取数据样本中的最大值，MIN——获取数据样本中的最小值.如果提供的计算操作参数不是这四个之一，客户端应该显示正确的错误消息。如果一切正常，服务器应该将计算结果发送到边缘设备（即客户端），并在终端正确显示结果。

DTE：删除数据文件

DTE fileID 边缘设备将在中央服务器端删除的特定文件的文件ID 作为参数包含在内，边缘设备（即客户端）收到此命令后，应向中央服务器发送消息以请求服务器删除参数中提供的具有该 fileID 的相应文件。服务器需要首先检查文件是否存在，如果文件不存在，服务器应该做出响应通知客户端要删除的文件不存在，客户端应该在终端显示适当的错误消息，例如，“该文件在服务器端不存在”。如果一切正常，服务器应该从其文件夹中完全删除文件。此外，中央服务器应维护一个名为delete-log.txt 的日志文件。如果一切正常，服务器应该附加一条删除操作日志消息，格式如下：

边缘设备名称；时间戳；文件ID； dataAmount 超级智能手表； 2022年9月30日10:33:13； 1个； 10 之后，中心服务器应向客户端响应消息，通知客户端文件已成功删除，客户端应显示成功消息（例如，“ID 为 fileID 的文件已成功从中心服务器删除”）。

AED：有源边缘设备

AED 此命令不应有任何参数。中央服务器应该检查除了发送 AED 命令的边缘设备之外是否还有其他活动的边缘设备。如果是这样，服务器应将边缘设备名称和自边缘设备加入以来的时间戳（及其 IP 地址和端口号，仅限 CSE 学生）从活动边缘设备日志文件发送到客户端（服务器应排除该信息向服务器发送 AED 命令的客户端）。客户端应在终端显示所有接收到的边缘设备的所有信息。如果没有其他活动边缘设备，则应向客户端发送“没有其他活动边缘设备”的通知消息并显示。接下来，客户端应提示选择可用命令之一。

6 OUT：退出边缘网络

出去

此命令不应有任何参数。客户端应关闭 TCP 连接，并在终端显示再见消息退出。服务器应更新其有关当前活动边缘设备和活动边缘设备日志文件的状态信息。即，根据来自客户端的消息（带有边缘设备名称信息），服务器应该删除该边缘设备，这需要删除活动边缘设备日志文件中包含该边缘设备的行（所有后续活动边缘设备在文件应该向上移动一行，并且它们的活动边缘设备序列号应该适当更新）并且应该向客户端发送确认并在终端上显示。请注意，不得删除此边缘设备上传的所有数据文件和消息。为简单起见，我们不会测试边缘设备忘记退出或退出不成功的情况。

3.4 点对点通信（视频文件上传，仅限 CSE 学生）

作业的 P2P 部分使一个边缘设备能够使用 UDP 将视频文件上传到另一台边缘设备（这种情况的一个很好的例子是摄像机上传其视频文件到智能手机）。每个边缘设备都处于两种状态之一，演示者或观众。 Presenter 边缘设备将视频文件发送到 Audience 边缘设备。这里，presenter 边缘设备是 UDP 客户端，而 Audience 边缘设备是 UDP 服务器。 Audience 边缘设备接收到视频文件后，保存文件和 Presenter 的设备名称。请注意，边缘设备可以在 Presenter 或 Audience 状态下运行。

要实现此功能，您的客户端程序应支持以下命令，并提供两个示例视频文件以帮助实现和测试此命令。

UVF：上传视频文件 UVF deviceName 文件名 Audience 边缘设备和文件名应作为参数包含在内。您可以假设参数中包含的文件将在具有正确访问权限集（读取）的客户端的当前工作目录中可用。您不应该假设该文件将采用特定格式，即只假设它是一个二进制文件。 Presenter 边缘设备（例如 uavcamera）应检查 Audience 边缘设备（由设备名称参数指示，例如 weismartpone）是否处于活动状态（例如，通过发出命令 AED）。如果 wei-smartphone 未激活，Presenter 客户端应在提示 uav-camera 客户端时显示适当的错误消息（例如，wei-smartphone 离线）。如果 wei-smartphone 处于活动状态，则 uav-camera 应在通过 UDP 将文件内容传输到 wei-smartphone 之前获取 wei-smartphone 地址和 UDP 服务器端口号（例如，通过发出命令 AED）。这里，uav-camera 是 UDP 客户端，wei-smartphone 是 UDP 服务器。该文件应存储在wei-smartphone当前工作目录中，文件名为presenter deviceName\_filename（请勿在名称中添加扩展名。如果文件名具有扩展名mp4，例如test.mp4应存储为uav-camera\_test .mp4 在我们的示例中）。文件名区分大小写，一个字长。文件传输完成后，uav-camera 的终端会提示选择一个可用的命令。 wei-smartphone 的终端应显示适当的消息，例如，已从 uav-camera 收到文件 (test.mp4)，然后提示选择可用命令之一。

测试注意事项： 1) 当你测试你的作业时，你可以在不同终端上的同一台机器上运行服务器程序和多个客户端程序。在这种情况下，使用 127.0.0.1（本地主机）作为目标（例如，我们上面示例中的 wei-smartphone）IP 地址。 2) 为简单起见，我们将在不同的目录下运行不同的客户端，并且不会测试客户端键入/发出命令时收到文件的场景，即接收客户端不需要通知文件转移到他们发出下一个命令之后。

3.5 文件名和执行

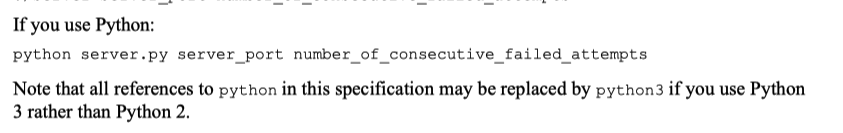
服务器和客户端（即边缘设备）的主要代码应包含在以下文件中：server.c 或 Server.java 或 server.py，以及 client.c 或 Client.java 或 client .py。您可以自由创建附加文件，例如头文件或其他类文件，并根据需要命名它们。

服务器应接受以下两个参数：

• server\_port：这是服务器将用于与边缘设备（即客户端）通信的端口号。回想一下，TCP 套接字不是由服务器端口号唯一标识的。因此，多个 TCP 连接可以使用相同的服务器端端口号。

• number\_of\_consecutive\_failed\_attempts：这是边缘设备应被阻止 10 秒之前连续不成功的身份验证尝试的次数。它应该是 1 到 5 之间的整数。

服务器应该在任何客户端之前执行。应按以下方式启动：



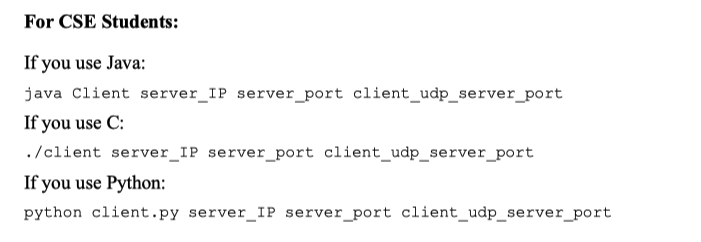
客户端应接受以下三个参数：

• server\_IP：这是运行服务器的机器的IP 地址。

• server\_port：这是服务器使用的端口号。此参数应与服务器的第一个参数相同。

• client\_udp\_port：这是客户端将侦听/等待来自其他客户端的UDP 流量的端口号。

请注意，您不必指定客户端使用的 TCP 端口。您应该允许操作系统选择一个随机的可用端口。同样，您应该允许操作系统为 UDP 客户端选择一个随机可用的 UDP 源端口。每个客户端都应在单独的终端中启动，如下所示：



注：1) CSE 学生用于 P2P UDP 通信的 client\_udp\_server\_port 附加参数在第 3.4 节中描述。在 UDP P2P 通信中，一个客户端程序（即 Audience）充当 UDP 服务器，另一个客户端程序（即 Presenter）充当 UDP 客户端。 2)

当您测试您的作业时，您可以在不同终端上的同一台机器上运行服务器和多个客户端。在这种情况下，使用 127.0.0.1（本地主机）作为服务器 IP 地址。

## 3.6 程序设计考虑

客户端程序设计 客户端程序应该相当简单。客户端需要通过命令行界面与人交互并打印有意义的消息。第 8 节提供了一些示例。您不必使用示例中显示的完全相同的文本。启动后，客户端应与服务器建立 TCP 连接并执行身份验证过程以加入边缘网络。身份验证后，应提示客户端输入可用命令之一。几乎所有命令都需要客户端与服务器之间的简单请求/响应交互。

对于 CSE 学生，客户端程序还涉及使用 UDP 的 P2P 通信。与上述类似，应提示用户输入可用的 P2P 通信命令：UVF。此功能应使用新线程实现，因为客户端程序在文件上传时可能需要运行其他命令。上传完成后线程将结束。同样，客户端 UDP 服务器应该使用另一个线程来实现。然而，这个线程应该一直运行到客户端离线，因为它是一个 UDP 服务器线程。您应该特别注意多个线程将如何与各种数据结构进行交互。课程网页上提供了所有受支持语言的多线程代码片段。

服务器程序设计 当服务器启动时，您可以假设没有活动的边缘设备。服务器应等待边缘设备连接，执行身份验证，并按顺序为客户端发出的每个命令提供服务。请注意，您需要定义多个数据结构来管理边缘网络的当前状态（例如，活动边缘设备和数据文件），并且服务器必须能够同时与多个边缘设备交互。实现此目的的一种可靠方法是使用多线程。在这种方法中，您将需要主线程来侦听新连接。这可以在 while 循环中使用套接字接受函数来完成。这个主线程是你的主程序。对于每个连接的边缘设备/客户端，您需要创建一个新线程。当与一个特定的客户端交互时，服务器应该接收到一个特定操作的请求，采取必要的行动并相应地响应客户端并等待下一个请求。您可以假设与客户端的每次交互都是原子的。考虑客户端 A 发起与服务器的交互（即命令）。当服务器在处理这个交互时，它不能被另一个客户端 B 的命令打断。客户端 B 的命令将在客户端 A 的命令被处理后执行。一旦客户端退出，相应的线程也应该终止。您应该特别注意多个线程将如何与各种数据结构进行交互。课程网页上提供了所有受支持语言的多线程代码片段。

# 4. 附加说明

• 这不是小组作业。您需要单独处理此问题。

• 入门提示：处理复杂实施任务的最佳方式是分阶段进行。一个好的起点是实现允许单个边缘设备登录服务器的功能。接下来，为几次不成功的尝试添加阻止功能。然后扩展它以处理多个客户端。一旦您的服务器可以支持多个客户端，

12 实现与服务器交互的功能。请注意，这可能需要更改您已经实现的某些功能的实现。一旦与服务器的通信正常工作，您就可以继续进行点对点通信（仅限 CSE 学生）。您必须严格测试您的代码，以确保所有可能的（和逻辑的）交互都可以正确执行。

• 应用层协议：记住您正在为边缘网络实施应用层协议。我们只考虑最终结果，即上述功能。您可能希望重新访问我们研究过的一些应用层协议（HTTP、SMTP 等），以查看消息格式、所采取的操作等的示例。

• 传输层协议：您应该使用 TCP 进行每个客户端和服务器之间的通信，（使用 UDP 进行两个客户端之间的 P2P 通信，仅限 CSE 学生）。 TCP 连接应由客户端在身份验证阶段建立，并应保持活动状态，直到边缘设备退出，而 UDP 则没有这样的要求。服务器的服务器端口被指定为命令行参数。 （类似地，UDP 的服务器端口号仅指定为客户端 CSE 学生的命令参数）。不需要指定 TCP 和 UDP 的客户端端口。您的客户端程序应该让操作系统选择任何可用的 TCP 或 UDP 端口。

• 备份和版本控制：我们强烈建议您经常备份您的程序。 CSE 每晚备份所有用户帐户。如果您在个人机器上开发代码，强烈建议您进行每日备份。我们还建议使用良好的版本控制系统，例如 Github 或 bitbucket，以便您可以回滚并从任何无意的更改中恢复。但是，不要将您的代码发布到公共存储库。有许多易于使用的服务。由于计算机故障、文件丢失等问题，我们不会接受任何特殊考虑的请求。

• 语言和平台：您可以自由使用C、Java 或Python 来完成这项任务。请选择您喜欢的语言。这些程序将在 CSE Linux 机器上进行测试。因此，请确保您的整个应用程序在这些机器（即 CSE 实验室计算机）或使用 VLAB 上正确运行。如果您计划在个人计算机（可能使用不同的操作系统或版本或 IDE）上开发和测试程序，这一点尤其重要。请注意，CSE 机器支持以下版本：gcc 版本 10.2、Java 11、Python 2.7 或 3.9。如果您使用 Python，请在您的报告中明确提及我们应该使用哪个版本的 Python 来测试您的代码。您只能使用您选择的编程语言中提供的基本套接字编程 API。您不得使用任何特殊的即用型库或 API 来为您实现规范的某些功能。

• 如第 8 节中的示例所示，终端上显示给用户的各种消息不要求您必须使用相同的文本。但是，请确保文本是明确的。

• 我们鼓励您使用 ED 论坛提出问题并讨论解决问题的不同方法。但是，您不应在论坛上发布您的解决方案或任何代码片段。

• 如果需要，我们将在第 7 - 9 周安排额外的咨询时间，以帮助您解决与作业相关的问题。