

机载网络数据时间同步处理算法的设计与实现

王 灏, 李 凯, 聂瑶佳

(中国飞行试验研究院 测试所, 西安 710089)

摘要: 数据准确可靠是飞行试验数据处理的基本要求, 但网络化测试架构的使用给数据处理时间同步带来一定的难度; 针对机载网络数据处理中存在的网络包乱序、时间不同步的问题, 研究了网络包数据乱序的产生原因及其特点, 详细分析并比较了基于包个数、基于最大逆序、基于半包时间间隔三种时间同步处理算法的优缺点及适用性, 最终选择使用基于半包时间间隔的时间同步处理算法作为机载网络数据处理算法, 并详细介绍了该算法的具体实现方式; 在实验中测试了该算法的有效性, 结果数据表明该算法能有效解决网络数据时间同步的问题并得到准确的处理结果, 保证了型号数据处理的准确性。

关键词: 网络化测试架构; 数据包乱序; 时间同步算法; 半包时间间隔

Design and Implementation of Time Synchronization Processing Algorithm for Airborne Network Data

Wang Hao, Li Kai, Nie Yaojia

(Chinese Flight Test Establishment, Xi'an 710089, China)

Abstract: Accurate data is the basic requirement for flight test data processing, but the use of networked test architecture brings some difficulty to data processing of time synchronization. Aiming at the problem of network packet out-of-order and time-synchronization in airborne network data processing, the causes and characteristics of network packet data out-of-order are studied. The number of packets based on the number of packets, based on the maximum reverse order, based on the half is analyzed and compared in detail. The advantages and disadvantages and applicability of three time synchronization processing algorithms in packet time interval are selected. The time synchronization processing algorithm based on half packet time interval is selected as the airborne network data processing algorithm, and the specific implementation of the algorithm is introduced in detail. The effectiveness of the algorithm is tested in the experiment. The data show that the algorithm can effectively solve the problem of network data time synchronization and obtain accurate processing results, which ensures the accuracy of model data processing.

Keywords: Networked test architecture; packet out-of-order; time synchronization algorithm; half-packet time interval

0 引言

当前飞行试验中广泛使用网络化测试架构, 网络化架构下采集的参数量大、采样率高且测试设备可采用成熟的货架产品^[1-3]。在网络化架构中采集器将采集的数据以网络包的形式发送至测试系统, 课题需要处理的参数就分布在这些网络包中。数据在测试系统上传输时可通过交换机方便地到达任何连接在交换机上的设备^[4-5]。由于数据的发送、传输、接收的过程中会存在一定的延时, 数据包在处理时就不会按照正常的时序排列, 存在数据包乱序的现象。如果在数据处理的过程中不对数据进行时间同步处理, 结果数据的时刻会发生错乱, 误把其他时刻的数据当做当刻数据进行处理, 就会导致结果数据不够精确, 无法保证型号数据处理的准确性^[6-7]。所以需要针对网络数据处理中存在的问题, 研究网络包数据乱序产生的原因及特点, 设计

并实现数据时间同步处理算法, 解决网络包数据乱序问题, 为课题提供精确的试飞数据。

1 网络包数据乱序原因及其特点

1.1 网络化测试系统简介

网络化机载测试系统包括传感器、采集器、交换机、记录器、数据处理子系统、遥测发射机等设备^[8]。采集器按照设定的频率采集模拟量、数字量、总线数据、视频数据等被测信号, 然后将测得的信号打包成标准格式的网络包数据发送, 网络包数据经过交换机发送至数据处理子系统、记录器、遥测发射机等设备。记录器记录接收到的网络包数据, 遥测发射机将数据通过 PCM 的形式发送至地面, 数据处理设备对接收到的网络包数据进行处理并存储处理结果^[9-10]。机载网络化测试架构如图 1 所示。

网络化机载测试系统具有如下几个特点:

1) 测试系统中一般有多台采集器, 每台采集器都会发送多条网络数据流, 数据流的目的地址及端口可以通过机载信息配置文件获取。

收稿日期: 2019-11-25; 修回日期: 2019-12-13。

作者简介: 王 灏(1992-), 男, 陕西渭南人, 助理工程师, 主要从事飞行试验实时监控与数据处理技术方向的研究。

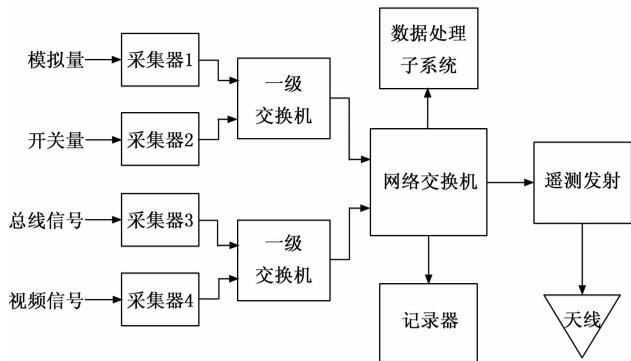


图 1 机载网络化测试架构

是未发生乱序的网络包，按照时间顺序依次排列；(b)是发生乱序的网络包，由于 B 包比 A 包延时大，所以 B 包总是排列在 A 包相应时刻之后。



图 2 数据包乱序示例

网络包数据乱序具有如下几个特点：

- 2) 每条数据流都有各自的采样率，各个流之间的采样率不完全一致，采集器按照该采样率发送数据。
- 3) 每条网络流的网络包都有各自独立的 key 值，该值是网络包的唯一标识。
- 4) 网络包数据中包含待处理参数的码值，通常情况下课题需要处理的参数分部在不同的网络包中。
- 5) 由于网络的复杂性以及传输的不确定性，接收到的网络包数据有可能存在丢包现象。
- 6) 数据处理时需要在不同采样率的不同网络数据流中进行码值提取和物理量转化，需要接解决时间同步问题。

1.2 网络包数据乱序原因及特点

由机载测试系统的架构可知，数据处理子系统通过交换机接收采集器发送的网络包数据，然后对数据进行处理。由于数据包的发送、传输、接收过程都会对数据包产生延时，而延时后的数据在处理时就不会按照预定的顺序进行排列，这就导致了数据包之间产生乱序，所以延时是数据包乱序的根本原因。延时由三部分组成：发送延时、传输延时、接收延时，网络包总延时等于这三个延时之和^[11-12]。对于发送、传输、接收的延时原因分析如下：

对于发送延时：1) 由于采集器设备性能的差异，不同采集器从数据产生到数据发送所用的时间不一样；2) 对于同一采集器的采样率相同的两个数据包，在发送同一时刻数据的时候会有先后顺序。

对于传输延时：1) 不同数据包的传输路径的长短不完全一致，例如位于垂尾的采集器和位于机身的采集器产生的数据到数据接收设备的路径长度显然不同。2) 网络包从产生到接收所经过的系统节点数不完全一致，例如有些经过一级交换机到核心交换机，有些会直接进入核心交换机，这些网络节点对数据包的产生的延迟不完全一样。

对于接收延时：1) 接收设备的性能会影响数据从到达接收设备到开始处理的时间。2) 设备在接收某一个数据包时，另一个数据包到来等待接收，会对下一个包产生接收延时。

假设系统中有 A、B 两个数据流，采样率都是 32，未发生延时和发生延时的数据包顺序如图 2 所示。其中 (a)

1) 单条流的网络包数据不会产生乱序，因为单条数据流的数据是依次发送的。乱序主要体现在数据流与数据流之间，即流与流之间会产生乱序。

2) 网络数据流的乱序规律是一定的，也就是说对于确定的测试系统，数据流的乱序是按照一定的乱序规律重复出现的。

由于网络包数据乱序，对处理时带来的困难是：无法按照数据包的接收顺序依次处理网络包数据，否则就会把其他时刻的数据误处理成当刻数据，所得的数据不够精确^[13-14]。所以需要针对数据包的乱序问题，设计一套时间同步算法，使得数据处理的结果在时间上能够对齐，保证试飞数据准确可靠。

2 三种时间同步算法的分析比较

针对机载网络化架构下数据处理时存在的网络包乱序问题，提出三种时间同步处理算法，并详细比较三种算法的优缺点，最终选择其中的一种作为机载数据处理算法进行实现。三种时间同步处理算法的详细描述如下。

2.1 基于包个数的时间同步处理算法

由于机载测试系统中每条网络流的数据包的采样率是固定的，同时课题需求的采样率也是固定的，所以可以根据网络包的采样率除以课题需求的采样率，得到网络包每秒需要处理的个数，实际计算时对网络包数据按照处理个数进行等间隔取数。基于半包间隔的时间同步算法的具体实现方式是：

1) 分析机载信息配置文件，获取每个网络包的 key 值及采样率；

2) 分析待处理的参数，找出包含待处理参数的网络包的 key 值及采样率；假设待处理网络包共有 N 个，采样率分别为 S1、S2、…、Sn；

3) 根据课题需求的采样率 S，计算每个包每秒计算的次数 $X_i = S_i / S$ ($i = 1, 2, \dots, n$)。

4) 若 $X_i < 1$ ，则令 $X_i = 1$ ，这是因为实际需求的采样率大于网络包的采样率时，对于每一个数据包都要进行计算。对数据进行解算时，网络包每 X_i 个计算一次；

5) 读取网络包数据并分析 key 值以及计算次数，若 key 值属于待处理参数的 key 值，并且计算次数能够整除

X_i , 则对该网络包进行处理, 否则不处理并读取下一个网络包; 依次循环往复;

6) 当每个网络包都处理了一遍时, 将当即网络包时刻作为处理时刻, 并将处理结果写入结果文件。

这种时间同步处理算法的优点是实现简单, 缺点是对网络数据的质量要求较高, 不能有乱序或者丢包。若测试系统的网络包数据没有乱序且不存在丢包现象, 则可以使用该算法, 但若存在丢包或乱序, 使用该算法处理的结果数据的时间就会存在偏差, 丢包或乱序越严重, 时间偏差就会越大。

2.2 基于最大逆序的时间同步处理算法

对于特定的机载网络化测试系统, 网络包的乱序是固定的, 只要找到乱序的规律, 再合理利用数据包乱序的规律, 就可以在处理时解决乱序问题实现时间同步。该算法的核心内容是: 1) 找到数据包乱序的规律, 即最大逆序时间, 2) 合理利用最大逆序时间实现时间同步。

网络中 1 秒接到的数据包数量为 N , 他们的时间分别为 T_1, T_2, \dots, T_n , 计算每两个数据包之间的时间差值 $\Delta T_i = T_{i+1} - T_i (i = 1, 2, \dots, N-1)$, 比较每个 ΔT_i , 得到最大值 ΔT_{max} , 该值就是最大逆序时间, 乱序的规律就以此逆序时间为周期反复出现的。假设数据中有 A、B 两种包, 采样率都是每秒钟 64, 由于数据乱序的原因, B 包总是比 A 包晚到 0.015625 (1/64) 秒, 即 $= 0.015625$, 数据在缓存中排列方式如表 1 所示。

表 1 数据排列

包到达时刻	A 包时标	B 包时标
0.000000	0.000000 A1	
0.015625	0.015625 A2	0.000000 B1
0.031250	0.031250 A3	0.015625 B2
0.046875	0.046875 A4	0.031250 B3
.....

表中每一行代表一个逆序时间周期, 每填满一个周期的缓存, 就可以正确处理一次网络包数据, 比如 0.015625 时刻收到的 A2 与 B1, 将数据暂时存放在缓存快 1, 0.031250 时刻收到的 A3 和 B2, 将数据放在缓存快 2, 这时可以将缓存快 1 的 A2 和缓存快 2 的 B2 结合就可以处理出来正确的参数。

基于最大逆序的时间同步处理算法的优点是可以通过找出最大逆序时间并合理利用乱序规律解决时间乱序的问题; 但缺点是该方式实现起来比较复杂, 同时对于网络数据的容错能力较弱, 假如数据包乱序较为严重, 找到最大逆序时间以及利用乱序规律都比较困难。

2.3 基于半包间隔的时间同步处理算法

由于网络包数据按照一定的频率发送至测试系统, 所以特定网络包不会发生乱序, 接收到的该网络包是以特定的时间间隔按序排列, 网络包时间间隔的一半称为半包间

隔。课题需求的采样率是确定的, 可以根据需求的采样率算出每秒中需要处理的时刻, 在这些时刻的左右半包时间间隔之内, 肯定有且只有一个网络包数据, 处理该网络包的数据就可得到该时刻的参数值。

例如网络中 A 包的采样率为 16, 需求参数的采样率为 8, 可以计算出 A 包的半包间隔 $halfPackGap = 1000/16/2 = 15.625$, 需求的采样率时刻的毫秒值分别为: 000、125、250、325、500、625、750、875。只要读取到的网络包时刻距这些时刻的时间差小于半包间隔就对数据包进行处理。基于半包间隔时间同步处理算法示例如图 3 所示, 图中上面的时刻表示需求的采样率时刻, 下面的时刻表示网络包数据真实采样率, 灰色部分表示需求采样率时刻附件的半包时间间隔, 若网络包时刻落在灰色部分就进行处理。

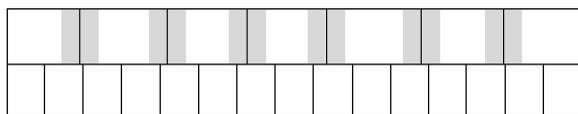


图 3 基于半包间隔时间同步处理算法示例

基于半包间隔的时间同步处理算法的优点是算法清晰明了且实现简单, 只需要对网络包中的时间与待处理时间进行比较就可以实现时间同步; 此外该算法的容错能力比较强, 若网络中存在丢包、乱序等现象, 由于进行了网络包的时间判断所以不会发生错乱, 更不会影响后面时刻数据包的处理。

由于测试系统中的丢包、乱序现象时有发生, 考虑到算法的易实现性、对数包的容错能力等因素, 选择基于半包半包间隔的时间同步处理算法作为机载网络数据时间同步处理算法。

3 基于半包间隔的时间同步处理算法的实现

利用基于半包间隔时间同步处理算法处理数据时, 按照一秒的数据量为单位进行处理, 该过程需要经过三个步骤: 设置数据缓冲区、处理数据、结果数据存储。数据缓冲区主要用于存储数据处理的中间结果, 避免频繁的 I/O 操作, 缓冲区的合理设置对于数据处理至关重要^[15]; 数据处理的过程就是按照基于半包时间间隔的数据处理算法处理网络包数据, 并将处理结果放入数据缓冲区中; 结果数据存储就是将缓存中存储的网络包数据写入到结果文件中。

3.1 设置数据缓冲区

为了存储数据处理的中间结果, 设置图 4 所示的缓冲区, 该缓冲区为一个数组, 用于存放一秒钟的数据处理结果, 前三个元素分别存放时、分、秒, 后面的元素按照参数的采样率的大小依次存放, 假如参数采样率为 8, 每 8 个元素存放一个参数一秒的结果, 这 8 个元素分别存放 000、125、250、325、500、625、750、875 时刻的处理结果。

设置三个这样的缓冲区, 分别用于存储上一秒、本秒、

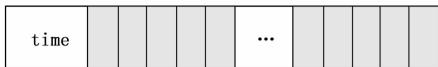


图 4 数据缓冲区的结构

下一秒的结果数据，这是因为网络数据中存在乱序现象，当本秒最开始的数据到来之后，有可能还会有上一秒最后时刻的数据；同样地当本秒的最后时刻到来之时，有可能下一秒最开始时刻的数据就会提前到来。设置三条这样的缓冲区就能够把所有的数据结果存储下来，不会存在结果数据丢失的情况。

3.2 数据处理方法

根据半包间隔时间同步的处理思想，设计基于半包间隔的时间同步处理算法的具体处理流程如下：

- 1) 分析机载信息配置文件，获取每个网络包的 key 值及采样率；
- 2) 分析待处理参数列表，在机载信息配置文件里找到待处理数据包的 key 值及采样率；
- 3) 若需要处理的网络包书数量为 N ，采样率分别为 $S_1、S_2、\dots、S_n$ ，计算每个数据包的半包时间间隔 $\Delta Ti = 1000/S_i/2$ 。
- 4) 根据待处理参数的采样率 K 计算每秒中要处理的时刻值， $T = \{T_0、T_1、\dots、T_{k-1}\}$ ，其中 $T_i = [1000/K] * i$ 。
- 5) 读取网络包数据，分析网络包数据的 K 值，若不需要处理则跳过读取下一个网络包，若需要处理，分析网络包的时间，判断时间属于本秒、上一秒还是下一秒时间，然后分析该时刻是否在需求采样时刻的半包时间间隔内，若不在则跳过读取下一个网络包，若是则判断该网络包属于第几个采样时刻，对网络包数据进行码值提取及物理量转化后，将值放在缓存中特定的位置。
- 6) 循环重复步骤 5) 的过程，当缓存满时将该缓存的数据写入到结果文件中。

3.3 结果数据存储

当缓存中的数据写满后，要将数据写入到结果文件中，然后对缓冲区进行清零并变更三个缓冲区的时间状态。一般情况才，上一秒的缓冲区最先被写满，当上一秒缓冲写满后，将该缓冲区的数据写入到结果文件中，然后对该缓冲区进行清零，接下来变更三个缓冲区的状态：上一秒缓冲接收下一秒数据，本秒缓冲接收上一秒数据，下一秒缓冲接收当刻数据，如此循环往复，保证了数据能够一直进行正确的缓存。

4 实验测试与验证

利用 C 语言实现了基于半包间隔的时间同步处理算法，在实验中测试该算法的有效性，需要验证两个方面的内容：一是数据结果的时间同步性，二是处理结果的准确性。

系统中存在 A、B、C 三个数据包，采样率分别为 16、32、64，三个数据包中分别有 $P_1、P_2、P_3$ 三个参数，处理

的采样率为 8，利用该算法进行两次数据处理：一是将三个参数放在一起一块处理，二是将三个参数单独各处理一次。进行两次处理的原因是：由于单个网络包不会存在乱序问题，单独处理某个特定的网络包，可以验证数据处理结果的准确性。三个参数一块处理所得结果如图 5 所示，三个参数单独处理所得结果的时间与数值和图 5 的时间与数值皆相同。

HH: MM: SS: MSS	P_1	P_2	P_3
09: 08: 12: 000	2. 350	3. 125	-1. 375
09: 08: 12: 125	2. 475	3. 264	-1. 422
09: 08: 12: 250	2. 600	3. 387	-1. 573
09: 08: 12: 375	2. 654	3. 422	-1. 632
09: 08: 12: 500	2. 725	3. 564	-1. 698
09: 08: 12: 625	2. 800	3. 633	-1. 712
09: 08: 12: 750	2. 850	3. 689	-1. 823
09: 08: 12: 875	2. 900	3. 722	-1. 875

图 5 数据处理结果

数据处理结果表明：单独处理时结果数据的时间与一块处理所得结果时间一致，验证了该算法解决了网络包乱序问题，保证了数据处理时间同步性。此外，单独处理时所得数据的数值和一块处理时的数值一致，验证了该算法能够得到准确的数据结果，保证了数据处理的准确定。

5 结束语

本文首先介绍了机载网络化测试架构并分析了网络包乱序产生原因及其特点，针对网络包乱序问题提出了三种时间同步处理算法，详细分析并比较了三种时间同步处理算法的内容及优缺点，最终选择基于半包间隔的时间同步处理算法作为机载实时数据处理算法，并给出了该算法的具体实现过程。试验结果表明该算法能有效解决机载网络数据处理时间同步问题，并给出精确的处理结果。

参考文献：

- [1] 左泽敏, 段小维, 王朝蓬. 空中飞行试验台新型网络化测试系统的设计 [J]. 航空科学技术, 2016, 27 (4): 50-53.
- [2] 杨廷梧, 田宝泉. 飞行试验新型遥测机载网络化采集与记录系统架构 [J]. 2013 (5): 59-63.
- [3] 滕飞. 基于网络化的机载测试系统关键技术研究 [D]. 西安: 西安电子科技大学, 2013.
- [4] 祁春, 聂瑶佳, 周训强. 嵌入式新型机载实时数据处理系统的设计与实现 [J]. 计算机测量与控制, 2018, 26 (3): 208-211.
- [5] 田方正, 王锦. 机载网络化测试系统框架设计技术研究 [J]. 测控技术, 2013, 32 (4): 9-11.
- [6] 王仲杰. 网络化机载测试系统的集成与测试 [J]. 中国测试, 2017 (3): 135-138.
- [7] 亓盛元. 基于 IEEE1588 时统的机载测试实时局域网设计 [D]. 西安: 西安电子科技大学, 2014.