

doi: 10.11823/j.issn.1674-5795.2018.02.13

飞行试验中 UDP 数据包丢失和失序处理研究

周训强

(中国飞行试验研究院, 陕西 西安 710089)

摘要: UDP 协议是飞行试验中机载网络化测试系统进行数据传输时采用的主要传输协议。与 TCP 协议相比, UDP 协议的主要特点是它是面向非连接的协议、网络开销低、传输速度快但传输不可靠、数据包会发生丢失和失序现象。由于处理飞行试验数据时必须考虑数据的时间相关性, 试飞数据处理软件需要开辟适当的缓存区来对数据按照时间进行重新排序。根据飞行试验中网络数据包时间标签具有周期性的特点, 给出了确定缓存区大小的公式, 并讨论了数据包丢失和失序情况下缓存区的重排算法。最后以真实飞行试验数据的处理结果说明了算法的成功应用。

关键词: UDP 协议; 飞行试验; 包丢失和失序; 时间相关性; 重排算法

中图分类号: V217

文献标志码: A

文章编号: 1674-5795(2018)02-0050-04

Research on UDP Packet Loss and Out of Order Processing in Flight Test

ZHOU Xunqiang

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

Abstract: UDP protocol is the main transmission protocol used in airborne test system for data transmission during flight test. Compared with the TCP protocol, the main feature of UDP protocol is that it is a connectionless oriented protocol, which has low network overhead, fast transmission speed unreliable transmission, and the data packet will be lost and out of order. Since flight test data processing must take into account the time correlation of data, the flight data processing software needs to develop an appropriate buffer to reorder the data in time. According to the periodicity of network packet time label in flight test, the formula of determining buffer area size is given, and the rearrangement algorithm of buffer is discussed in the case of packet loss and disorder. Finally, the processing result of real flight test data shows the successful application of the algorithm.

Key words: UDP transport; flight test; packet loss and out of order; time correlation; rearrangement algorithm

0 引言

随着我国航空工业的不断发展, 研制型号和试飞型号不断增加, 飞行试验中测试参数越来越多, 数据流量也越来越大, 使得集中式采集、分布式采集等传统的机载数据采集系统架构已经不能满足测试技术要求。2004 年 10 月由美国发起的 iNET 项目开发技术得到了广泛的支持并在数年内形成了比较成熟的技术框架和技术发展路线^[1]。在国内, 飞行试验测试领域一直跟踪 iNET 的发展, 于 2012 年提出了空天地一体化遥测网络系统^[2-3]。

得益于国内试飞测试技术的迅速发展, 基于以太网技术的高带宽多数据流传输和记录在飞行试验中得以广泛应用, 机载网络化数据采集记录系统的概念及相关产品应运而生。

机载网络化数据采集记录系统的一般架构为: 多

台采集器通过以太网挂接到网络交换机, 在交换机上进行数据包汇聚, 再由同样挂接在交换机上的网络化记录器对这些数据包进行接收记录。更为庞大的测试系统(采集终端数量多)甚至存在多级交换机的级联, 而这些设备采用统一的时间源进行时间同步^[4]。网络化测试系统的广泛应用使得基于以太网的高带宽多数据流的传输和分析成为了飞行试验领域的研究热点。

1 失序和丢包分析与处理

1.1 失序和丢包分析

在飞行试验领域, 目前以太网数据包的传输主要采用 TCP 和 UDP 协议。TCP 采用面向连接的数据管理策略并通过重传保证数据可靠传输, 多用于传输通信和控制数据包等可靠性要求高的场合。但对于高带宽数据, 频繁的确认和重传无法保证数据的实时传送要求, 因此一般采用无连接的 UDP 传输协议, 其目的是

减少不必要的网络开销,提高传输带宽和传输效率。正因为如此,UDP 传输协议不提供可靠性、流量控制和差错恢复的手段^[5]。

通常,在机载网络化测试系统中,数据从发送端到目的端要经过若干的中间节点,节点与节点之间对数据的传输会产生延时。由于不同数据包可能经过不同的中间节点到达目的端,因此总的延时也不同,这导致数据包到达目的端的顺序与期望顺序可能不同,即失序。此外,由于缓冲区的溢出,数据包可能被中间节点拒绝,使得数据包丢失^[6]。很显然,数据失序不满足试飞数据分析对数据时间相关性的严格要求。为了满足数据分析时的时间相关性要求,需要对发生失序的数据进行重排,对丢失的数据进行补点。为此,在期望数据包全部到达之前要将将来的数据包暂存于一个缓冲区内。由此带来的问题:缓存区的大小如何确定;暂存数据在缓冲区的存放规则是什么;发生丢包时如何处理。下文将对这些问题进行讨论和分析。

1.2 重排缓存区的确定

根据网络交换机对数据包的缓存及转发原理,在网络拓扑结构、网络中的设备不发生改变,以及数据包在网络按周期规律传输的情况下,逆序情况也是稳定的,从实际的机载网络化测试系统中的数据包样本的时间标签的统计计算结果来看,也证明了这一点。机载网络化测试系统中,每种网络数据包的时间周期都是毫秒级(1/1024 ms ~ 1/16 ms)的,对于一段完整的数据包样本,在考察其逆序情况时,只需分析其中 1 s 时间跨度内的数据包的逆序情况即可。

根据飞行试验中网络数据包传输大多具有周期性的特点,假设网络中的数据包的种类(一般由包识别字确定)为 n 种,它们的传输周期分别为 $T_1, T_2, T_3, \dots, T_n$ 。根据上文的论述,在发生失序的情况下,需要开辟缓存区来对失序的数据包进行重排,而对失序情况进行量化是确定缓存区大小的重要工作,其方法是在网络数据包发送的目的端对接收的网络数据包时间标签做统计计算,假设在 1 s 内接收到的数据包的个数为 N ,它们的时间标签分别为 $t_1, t_2, t_3, \dots, t_N$,对相邻的两个包的时间标签进行如下计算

$$\Delta t_i = t_{i+1} - t_i \quad (i=1, 2, 3, \dots, N)$$

式中: Δt_i 为逆序时间,在 $\Delta t_i < 0$ 情况下,对 $|\Delta t_i|$ 求最大值 $\Delta t_{\max}$, $\Delta t_{\max}$ 为最大逆序时间。

逆序时间是对机载网络化测试系统中数据包逆序情况的量化描述,求出一个满足如下条件的最小值 T 。① T 为 $T_1, T_2, T_3, \dots, T_n$ 的最小公倍数;② $T \geq \Delta t_{\max}$ 。

那么申请 2 个大小为 $L = \frac{T}{T_1} \cdot L_1 + \frac{T}{T_2} \cdot L_2 + \dots + \frac{T}{T_n} \cdot L_n$ 的缓存,其中 $L_i (i=1, 2, 3, \dots, n)$ 为第 i 种数据包的大小,就可以对接收到的网络数据包进行排序。

需要说明的是,之所以采用 2 个大小为 L 的缓存区来对数据包进行排序,而不是采用更多个数的缓存区,是考虑到尽量减小因排序带来的时间延时。

1.3 排序算法的分析与实现

以实际的飞行试验数据为例,进行排序算法的分析。现有某型号某架次飞行试验数据的 A, B, C 三种网络数据包,周期分别为 1, 2, 4 ms,那么顺序情况下,数据包的接收情况应如表 1 所示。

表 1 顺序情况下的数据包接收情况

时间/ms	包 A	包 B	包 C
0	A ₀	B ₀	C ₀
1	A ₁		
2	A ₂	B ₁	
3	A ₃		
4	A ₄	B ₂	C ₁
5	A ₅		
6	A ₆	B ₃	
7	A ₇		
...	...	...	...

缓冲区的数据包排列顺序为 A₀B₀C₀A₁A₂B₁A₃A₄B₂C₁A₅A₆B₃A₇,记为顺序 X。而真实的情况是该数据样本有逆序情况发生,数据包的到达情况如表 2 所示。

表 2 逆序情况下的数据包接收情况

时间/ms	包 A	包 B	包 C
0	A ₀		
1	A ₁		
2	A ₂	B ₀	
3	A ₃		
4	A ₄	B ₁	
5	A ₅		
6	A ₆	B ₂	
7	A ₇		C ₀
8	A ₈	B ₃	
9	A ₉		
10	A ₁₀		
11	A ₁₁		C ₁
12	A ₁₂		
13	A ₁₃		
14	A ₁₄		
15	A ₁₅		C ₂
...	...	...	...

缓冲区的数据包排列顺序为 $A_0 A_1 A_2 B_0 A_3 A_4 B_1 A_5 A_6 B_2 A_7 C_0 A_8 B_3 A_9 A_{10} A_{11} C_1 A_{12} A_{13} A_{14} A_{15} C_2$, 记为顺序 Y , 经统计, 最大逆序周期为 7 ms。

当到达的数据包发生失序的情况时, 需要在保证到达的数据包能够全部缓存下来的前提下, 对数据包进行重新排序, 以达到时间对齐的目的。在发生丢包或者失序的情况下, 重新排序的过程就是将缓存区数据包的排序情况由 Y 变成 X 的过程。

按照上文对于缓存区大小确定方法的论述, $T = 8 \text{ ms}$, 可以申请 2 个大小为 $L = \frac{8}{1} \cdot L_A + \frac{8}{2} \cdot L_B + \dots + \frac{8}{4} \cdot L_C$ 的缓存区 H_1, H_2 , 按照表 1 中数据包排放规律来对到达的数据包进行缓存和排序。

当数据包按照表 1 中的规律到达时, 将数据包内的时间标签在 $0 \sim 7 \text{ ms}$ 的数据包按照表 1 的数据包排列规律放入缓存区 H_1 中, 将数据包内的时间标签在 $8 \sim 15 \text{ ms}$ 的数据包按照表 1 的数据包排列规律放入缓存区 H_2 中, 接下来, 在 16 ms 时刻, A_{16} 包到达, 根据最大逆序时间的概念, 表明 $0 \sim 7 \text{ ms}$ 时间段的数据包已经全部接收完毕(如果缓存区 H_1 仍未填满, 表明 $0 \sim 7 \text{ ms}$ 时间段的数据包有丢包发生), 此时可以将之前其它任务(数据处理任务或数据存储任务)中的数据存放缓存区清空, 然后将 A_{16} 按照表 1 中的数据包排放规律放入缓存区 H_1 。需要强调的是, 如果当前数据包较前一数据包的时间跨度大于 7 ms , 即大于 T , 表明已经开始接收到新的时间段的数据包, 此时需要考察数据包的时间标签 t_m 与缓存区内数据包的时间跨度(t_n, t_{n+7})的关系, 判断 t_m 是否属于 (t_n, t_{n+7}) 区间, 如果当前数据包不属于缓存区 H_1, H_2 中的任何一个, 则需要将缓存区 H_1, H_2 中时间靠前的缓存区进行数据包的完整性检查并对丢失的数据包按照相应的方法进行补齐, 接下来将缓存区提交至其它任务后清空, 然后将 A_{16} 按照表 1 的数据包排放规律放入缓存区。具体的算法流程图如图 1 所示。

采用 C++ 高级程序语言实现了上文所述算法^[7]。

2 算法应用

在实现主要算法的基础上, 完成了网数据处理软件的设计开发^[8], 为了验证排序算法的正确性, 使用本软件对两个不同型号飞机的多架次飞行数据进行处理。图 2 是某型号飞机某架次飞行记录的某时间段原

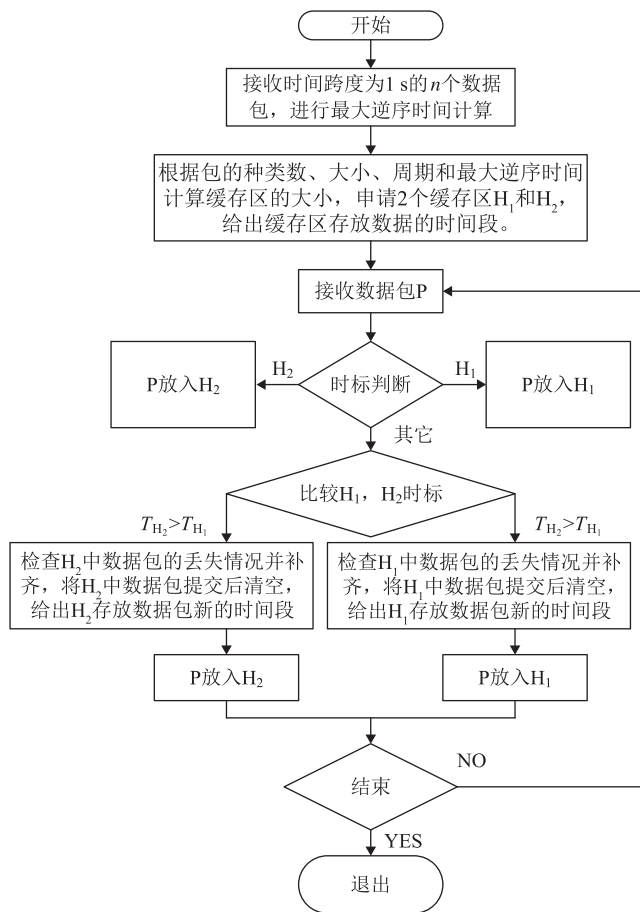


图 1 数据包排序算法流程图

键值	长度	时间	序号	键值	长度	时间	序号
6301	88	0:38:49:328.125000	14164	6301	88	0:38:49:328.125000	14164
6304	104	0:38:49:335.938000	30024	6302	104	0:38:49:328.125000	14164
6304	104	0:38:49:336.914000	30025	6402	296	0:38:49:328.125000	14164
6304	104	0:38:49:337.891000	30026	6303	136	0:38:49:328.125000	14164
6304	104	0:38:49:338.867000	30027	6304	104	0:38:49:335.938000	30024
6302	104	0:38:49:328.125000	14164	6304	104	0:38:49:336.914000	30025
6402	296	0:38:49:328.125000	14164	6304	104	0:38:49:337.891000	30026
6304	104	0:38:49:339.844000	30028	6304	104	0:38:49:338.867000	30027
6304	104	0:38:49:340.820000	30029	6304	104	0:38:49:339.844000	30028
6303	136	0:38:49:328.125000	14164	6304	104	0:38:49:340.820000	30029
6304	104	0:38:49:341.797000	30030	6304	104	0:38:49:341.797000	30030
6304	104	0:38:49:342.773000	30031	6304	104	0:38:49:342.773000	30031
7100	13	0:38:49:343.750000	14165	7100	13	0:38:49:343.750000	14165
6300	13	0:38:49:343.750000	14165	6300	13	0:38:49:343.750000	14165
6400	13	0:38:49:343.750000	14165	6400	13	0:38:49:343.750000	14165
6401	32	0:38:49:343.750000	14165	6401	32	0:38:49:343.750000	14165
6304	104	0:38:49:343.750000	30032	6301	88	0:38:49:343.750000	14165
6304	104	0:38:49:344.727000	30033	6304	104	0:38:49:343.750000	30032
6304	104	0:38:49:345.703000	30034	6304	104	0:38:49:344.727000	30033
6304	104	0:38:49:346.680000	30035	6304	104	0:38:49:345.703000	30034
6304	104	0:38:49:347.656000	30036	6304	104	0:38:49:346.680000	30035
6304	104	0:38:49:348.633000	30037	6304	104	0:38:49:347.656000	30036
6304	104	0:38:49:349.609000	30038	6304	104	0:38:49:348.633000	30037
6304	104	0:38:49:350.586000	30039	6304	104	0:38:49:349.609000	30038
6301	88	0:38:49:343.750000	14165	6304	104	0:38:49:350.586000	30039
6304	104	0:38:49:351.563000	30040	6304	104	0:38:49:351.563000	30040
6304	104	0:38:49:352.539000	30041	6304	104	0:38:49:352.539000	30041
6304	104	0:38:49:353.516000	30042	6304	104	0:38:49:353.516000	30042
6304	104	0:38:49:354.492000	30043	6304	104	0:38:49:354.492000	30043

排序前

排序后

图 2 排序前后数据包顺序对比图

始数据的相关信息。其中,键值表示包的识别字,序号表示该键值的包的当前个数。从图2中排序前的信息可以看出,数据包发生了失序。图2中的右图是排序后的数据包信息,可见已实现时间对齐。

本文提出的关于数据包的丢失和失序处理算法可以在飞行时间内的网络数据包实时接收时应用,以保证进入机载记录设备的数据包已经按照时间顺序进行存储。如果在飞行时间内由于机载测试系统软硬件资源的限制,不便对数据包进行实时排序,也可以在事后数据预处理环节中使用本算法。两种使用场合算法完全一致,无需作任何更改。

3 结论

通过对UDP传输协议和机载网络化测试系统数据包传输过程的分析,说明了数据包发生失序和丢失的原因。鉴于飞行试验数据处理分析对数据的时间相关性的严格要求,提出了开辟适当的缓存区来对失序的数据包按照时间排序的详细算法;之后,对缓存区大小的确定和排序算法进行了详细的论述,并在数据处理软件中实现了该算法的设计开发;最终的应用效果表明,该算法成功解决了飞行试验网络化测试系统中发生失序和丢包情况下的数据处理问题。

参 考 文 献

- [1] iNET system Architecture Version 2007.1 [EB/OL]. <http://www.irig106.org>.
- [2] 杨廷梧. 新型遥测系统中网络化测试技术展望[J]. 测控技术, 2010, 29(Z): 141-145.
- [3] 白效贤, 杨廷梧, 袁炳南. 航空飞行试验遥测技术发展趋势与对策[J]. 测控技术, 2010, 29(11): 6-9.
- [4] 杨廷梧, 王云山, 滕斌. 飞行试验遥测机载测试技术的发展与应用[J]. 测控技术, 2013, 32(4): 5-8.
- [5] Stevens W. R. TCP/IP 详解. 卷1, 协议[M]. 北京: 机械工业出版社, 2008.
- [6] 鲁宏伟. 基于UDP传输协议的包丢失和失序处理[J]. 计算机工程与应用, 2001, 37(2): 48-49.
- [7] 张永强(译). C++ 高级编程[M]. 北京: 清华大学出版社, 2015.
- [8] 陈坚. Visual C++ 网络高级编程[M]. 北京: 人民邮电出版社, 2001.

收稿日期: 2018-03-01



周训强(1983-), 男, 湖北荆门人, 工程师, 主要从事机载测试技术及飞行试验数据处理研究。

《宇航计测技术》2018年第2期目次

飞行器射前瞄准误差概率分析与控制

基于镁合金基体的螺纹连接承载性能研究

静压气浮轴承工程设计方法研究

基于光纤复用技术的弹体表面场参数组网测量技术

三点差分GPS定向方位引入方法研究

多通路激光火工品等效测量装置设计

基于Ansys Icepak的负载器热设计与热分析

拉扭疲劳试验机同步校准技术研究

倾斜摇摆试验台校准技术研究

一种冲击响应谱试验设备校准装置的研制

TTE/TTP技术在航天器中的应用分析

基于互相关算法的环境振动检测方法研究

基于ADE7858的交流电源监测模块设计

一种用于动态跟踪瞄准的高精度电控导轨设计与实现

基于微处理器的同步串行接口技术研究

结构健康监测技术在航天运载器结构中的应用探索

计量测试软件验证方法研究基于STM32的直流电机调速新方法

基于VB.NET的砣码自动校准系统软件设计与实现