文章编号:1000-582X(2010)08-029-07

多媒体文件传输嵌入式终端接收能力的匹配

王海威[1,2],倪　宏[2],孙　鹏[2],朱小勇[2]
(1. 中国科学技术大学 中科院网络传播系统与控制重点实验室,安徽 合肥 230027;
2. 中国科学院声学研究所 国家网络新媒体工程技术研究中心,北京 100190)

摘　要:提出了一种在网络带宽充足的条件下,能够匹配嵌入式终端接收能力的大型多媒体文件传输协议 EMFTP(embedded multimedia file transfer protocol,嵌入式媒体文件传输协议)。该协议是基于 UDP 的应用层传输协议,采用统一重传的差错恢复策略,并引入了基于带宽探测的递减式加增乘减速率控制算法,在尽可能的降低嵌入式接收端的资源占用前提下,提高数据传输速率。实验表明,与传统的 UDT 和 FTP 传输协议相比,EMFTP 的数据传输速度分别提高了 3%和 17%,接收方终端的 CPU 占用率分别降低 50.3%和 42.3%,内存占用率降低 15%和 4%。

关键词:网络协议;嵌入式终端;流量控制;差错恢复;带宽探测
中图分类号: TP393　　**文献标志码**:A

Embedded terminal receiving capability matching in multimedia file transmission

WANG Hai-wei[1,2], *NI Hong*[2], *SUN Peng*[2], *ZHU Xiao-yong*[2]
(1. Key Laboratory of Network Communication System and Control, Chinese Academy of Sciences; University of Science and Technology of China, Hefei 230027, Auhui, P. R. China;
2. National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, P. R. China)

Abstract: A novel transport protocol for large-scaled multimedia file - EMFTP (Embedded Multimedia File Transfer Protocol) is proposed to match the receiving capability of embedded terminal with sufficient network bandwidth. EMFTP is an application-level protocol using UDP, and adopts a method of uniform retransmission as its error recovery strategy and it also introduces an improved DAIMD (AIMD with decreasing increases) rate control algorithm based on bandwidth probe. It can improve the data transfer rate with a low resource-taken of embedded client. As the experimental result shows, compared with UDT and FTP protocol implemented on embedded terminal, EMFTP data transfer rate is increased by 3% and 17% respectively, the embedded client CPU occupancy ratios are reduced by 50.3% and 42.3% separately, and memory usage reduced by 15% and 4%.

Key words: network protocols; embedded terminal; flow control; error recovery; bandwidth probe

收稿日期:2010-04-01
基金项目:"十一五"国家科技支撑计划资助项目(2008BAH28B04)
作者简介:王海威(1984-),男,中国科技大学博士研究生,主要从事网络协议、嵌入式系统、多媒体通信等方向研究,(Tel)15811157152;(E-mail)wanghw@dsp.ac.cn。

“三网融合”的全面启动，带来的是网络业务和内容的不断融合和扩展，以数字机顶盒为代表的嵌入式终端，作为媒体服务的呈现载体，为用户访问网络，享受音视频媒体服务提供了一种有效的方式。针对网络中日益丰富的高清媒体资源，如何将其从服务器快速传送到嵌入式终端上，是媒体分发系统中的重要研究内容。随着网络技术的发展，网络带宽容量不断增大，渐渐不再成为文件传输的瓶颈。然而，由于嵌入式终端资源有限，网络处理能力较弱，较强的发送端发送能力和有限的接收端接收能力的不匹配，是影响文件传输速率的重要因素。另一方面，带宽增加带来的传输速率的提高，给资源受限的接收端带来了更大的压力。文件传输作为后台应用，要尽量减少资源使用量，不能影响媒体播放等前端业务的正常开展。因此在高带宽网络中，需要一种轻型的文件传输协议，用以匹配收发端的传输能力，降低接收端的资源压力。

传统文件传输协议，如改进的 TCP 拥塞控制算法[1-4] 和基于 UDP 的应用层方案 TFTP[5]、RBUDP[6-7]、UDT[8]等，都是针对普通计算机间的文件传输设计，没有考虑针对接收方接收能力和资源限制的优化。在嵌入式终端上运行时，CPU 和内存使用率过高，系统开销较大。

在嵌入式系统领域，为了降低系统开销，一些研究者提出了嵌入式协议栈，如 lwIP[9] 和 NanoIP[10]，通过对传统 TCP/IP 协议栈的修改和裁剪，满足嵌入式系统在功耗、成本等方面的要求，提供可靠的文件传输。但是，这种方法需要根据特定系统和应用需求进行有针对性的裁剪和设计优化，没有通用标准，降低了灵活性和可移植性。

文献[11]提出了一种应用于无线通信网络的嵌入式可靠传输协议 ERT。它的传输对象是误码率高的无线网络环境中短而零星的报文，而且速率控制机制过于简单，不适用于高速网络大型媒体文件的传输。

由于目前并没有完全符合嵌入式系统大型媒体文件传输需求的应用层协议出现，因此提出一种旨在匹配收发端传输能力，降低接收端的资源消耗的嵌入式系统多媒体文件传输协议 EMFTP。该协议是基于 UDP 的应用层传输协议，采用统一重传的差错恢复策略和基于带宽探测的递减式加增乘减速率控制算法[12]，在尽可能降低嵌入式接收端资源占用的前提下，保证数据传输速率。

1 EMFTP 的设计

1.1 体系结构

EMFTP 是集 TCP 协议和 UDP 协议于一体的应用层传输协议，使用 UDP 协议进行数据传输和应答反馈，TCP 协议进行请求丢包重传。数据传输中，发送方的功能包括数据发送、速率控制和丢包重传，而接收方的相应功能分别为数据接收、应答反馈和丢包维护。发送方将媒体文件分割成数据块，加上序列号和时间戳后封装到大小固定的 UDP 数据包（当前版本为 1 480 字节）中，使用带宽探测计算发送速率，并发送数据包。接收方统计丢包率情况，并计算和更新可用带宽，定期反馈给发送方供其调整发送速率。文件传输过程分为传输期和重传请求期 2 个阶段交替进行。在传输期，发送方将接收方还没有正确接收的所有数据包依次发送给接收方，传输完毕进入重传请求期，接收方将丢失的数据包列表反馈给发送方。这个过程反复进行，直到所有的数据包都被成功接收。图 1 描述了 EMFTP 的体系结构。

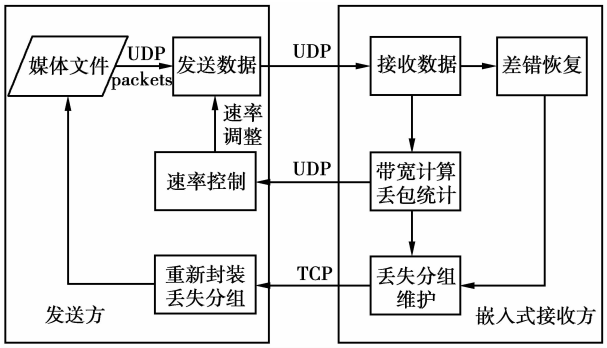

图 1 EMFTP 的体系结构

1.2 差错恢复

数据传输的可靠性通常通过回退 N 步或选择性重传来保障。这 2 种方法都需要接收方频繁的反馈，增加了嵌入式终端的工作强度，会引起终端数据接收处理能力的下降。因此，EMFTP 采用一种统一重传的差错恢复机制，简化接收端的应答，降低终端资源负担。

在第一个传输期，发送方将整个文件通过 UDP 协议发送给终端。由于 UDP 提供的是不可靠的数据传输，存在丢包的可能性，因此接收方校验收到的每个数据包的序列号，如果发现丢包，则记录下丢失的包号。如果一次丢包事件（丢包事件是指丢失一个或几个连续数据包的事件）内连续丢失了多个数据包，则记录丢失的第一个和最后一个数据包的序列号，并加上连续丢包的标志位，以降低反馈的数据量。传输期结束时，发送方通过 TCP 向接收方发送传输完成的信令，进入重传请求期，接收方将丢包链表反馈给发送方，发送方在下一个传输期重传丢失

的数据包。如果丢包链表为空，说明接收方已经成功接收整个文件，向发送方发送传输完成的信令。整个过程循环进行，直到接收方成功接收到所有的数据包。

具体算法如下。

算法 1：差错恢复。

发送方

1）接受终端文件下载的请求，读取媒体文件数据；

2）进入传输期，将媒体数据封装为 EMFTP 规定格式的数据包，发送给终端；

3）接收终端的反馈信令，根据丢包率和可用带宽进行速率控制；

4）发送数据完毕，向终端发送传输完毕的控制信令“FIN”；

5）进入重传请求期，接收终端的重传请求。如果收到丢包列表，重传丢失的数据包，转 2），如果收到终端传输完成的信令“FIN”，则关闭连接，传输完成。

接收方（终端）

1）终端向发送方请求文件传输；

2）进入传输期，接收分组，校验序列号，判断数据包是否丢失，并将丢失的包号记录进丢包列表，丢失的数据包在缓冲中以空包代替；

3）统计丢包情况，计算可用带宽，反馈给接收方；

4）接收缓冲区达到一定程度后，写入磁盘；

5）收到发送方的文件传输完成“FIN”信令，进入重传请求期，如果丢包列表不为空，将丢包列表反馈给发送方，请求重传，转 2），否则，向发送方发送结束信令“FIN”，关闭连接。

1.3　速率控制

在嵌入式终端接收能力的限制下，为了有效减少数据包的丢失，提高文件传输的效率，必须采用可靠高效的速率控制策略，及时跟踪终端负载状态的变化，调整发送速率以适配终端的接收能力。

速率控制算法采用基于探测的 DAIMD[12] 算法。

1.3.1　DAIMD

在 DAIMD 算法中，速率的增量 $\alpha(x)$ 是当前发送速率的非增函数，随着发送速率的增长，速率的增量逐渐趋近于 0。在需要降速时，发送速率以一个固定的降速因子 $\beta(0<\beta<1)$ 按比例减小。即增速函数和降速函数分别为

$$x = x + \alpha(x), \tag{1}$$

$$x = (1-\beta)\cdot x, \tag{2}$$

文献[12]证明了这种机制的公平性。

1.3.2　算法描述

EMFTP 采用统一重传的差错恢复策略，不需要对传输过程中的丢包进行应答和重传，但接收方仍然需要定期发送应答包，将数据接收情况反馈给发送方进行速率控制。

应答包内包括如下信息：1）该反馈间隔内收到的数据包总数 totalRecv；2）该反馈间隔内丢失的数据包数目 totalLost；3）探测得到的网络带宽 B；4）该反馈间隔内发生的丢包事件个数 LostEventNum。其中，totalRecv 和 totalLost 用来计算反馈间隔内的包丢失率 P，它代表最近一段时间内，丢包的严重程度；网络带宽 B 用来调整速率增量。

发送方每次收到应答包后，根据 totalRecv 和 totalLost 计算前一个反馈间隔内的丢包率 P

$$P = \text{totalLost}/(\text{totalRecv} + \text{totalLost})。 \tag{3}$$

根据丢包率的大小，分为 3 种情况讨论

1）$P=0$。没有丢包，说明网络状况良好，终端的接收能力也能够满足发送速率的要求，因此可以使用公式(1)进一步增大发送速率。

EMFTP 采用了一个与网络带宽相关的分段函数作为增速增量 $\alpha(x)$，该函数在发送速率 x 增大时快速的减小

$$\alpha(x) = 10^{\lfloor \log(B-R(x)) \rfloor - \tau} \cdot \frac{1}{\text{SYN}}, \tag{4}$$

其中：x 的单位是 dB/s；SYN=0.01 s 是接收方的定期反馈间隔，B 是探测得到网络带宽，单位为 bps；$R(x)$ 是以位/s 为单位的当前传输速率，即 $R(x)=8x$；τ 为增速控制因子，其取值控制着 $\alpha(x)$ 的变化幅度，采用经验值 3。

2）$0<P<0.05$。存在少量丢包，说明网络负载较高，或达到了终端处理能力的上限。对于文件传输来说，最重要的是提高数据的有效吞吐量，轻微的丢包可以获得更大的有效吞吐量，因此可以允许一定量的丢包来获取有效吞吐量的提升（丢失的包通过传输期结束时的差错重传机制来恢复）。因此，在这个阶段，发送方维持原先的速度继续发送数据。

3）$P>0.05$。丢包严重，说明此时网络已严重过载，或者发送速率已经超过终端接收能力的上限，这时，过大的发送速率不仅无法提高传输效率，反而会消耗终端更多的资源以维护丢失数据包的信息，因此，需要降低发送速率。然而，网络或者终端偶尔的突发事件或异常也可能会引起终端短期内丢包率的突增，这种情况下，是不应该降速的。因此，采用延期降速的策略，当首次发现丢包率超过阈值时，保持发送速率不变，当连续 2 个应答包的丢包率都超过阈值，才使用降速策略降速。

如果网络中有多个数据流同时检测到拥塞情况并降低发送速率，总体吞吐量会产生很大的振荡，导致带宽使用量的降低。采用一种随机降速的策略避免这种情况的出现。假设连续丢包的几个反馈间隔内的平均丢包事件数为 M，整数 N 服从 1 到 M 的均匀分布，则发送速率下降为

$$x = (1-\beta)^N \cdot x。 \tag{5}$$

降速因子 β 的选择十分重要。如果过小，不能及时降低速度以跟踪网络和终端负载的状态变化；如果过大，则会降低传输性能，并导致传输速度过于振荡。采用经验值 0.1。

由于应答包存在丢失的可能性，因此，如果接收方连续 4 个反馈周期内没有收到应答包，则认为发生了一个丢包事件，按照降速方法降低发送速率。

具体算法描述如下。

速率控制算法：

1)if received ACK

a) P=totalLost/(totalRecv+totalLost)；

b) if P==0,then

$x=x+\alpha(x)$；

c) else if 0<P<0.05

x=x。

d) else

i. if is_firsttime

x=x；

avgLEN=LEN。

ii. else

avgLEN=(avgLEN·7+LEN)/8；

N=random(avgLEN)；

$x=(1-\beta)^N \cdot x$。

e) Reset timer。

2) if timeout>SYN4

a) $x=(1-\beta)\cdot x$；

b) Reset timer。

3) Update timer, go to 1。

其中，LEN 和 avgLEN 分别为当前反馈周期丢包事件数和平均丢包事件数。

1.3.3 带宽探测

EMFTP 使用包对法[13]进行链路带宽探测。发送方每隔 16 个数据包连续发送一对没有时间间隔的包(称作一个包对)，接收方记录包对中两个数据包到达的间隔时间 intertime，通过中值过滤器过滤掉过大和过小的值后，使用平均值估计带宽容量。设数据包的大小为 S，平均间隔时间为 T，则带宽可以被估计为 S/T。

使用包对法进行带宽估计可能会受到其它因素的影响。如网络中存在其他数据流可能会导致带宽容量的低谷；而有些网卡使用的中断合并处理机制将会高估带宽容量。这些干扰可以通过使用多个包对的平均间隔时间降低或消除。为了降低带宽估计误差对速度控制产生的影响，EMFTP 的速度增量函数 $\alpha(x)$使用了上界函数来屏蔽误差。

2 实验结果

设计了 4 组实验，分别从传输性能、资源消耗、业务共存性和公平性 4 个方面分析 EMFTP 的性能，验证 EMFTP 对终端接收能力的适配性。

实验的 Server 节点和 Client 节点间的端到端有效网络带宽为 100 Mbps，平均往返时延(RTT)为 0.4 ms，实验中网络上没有其他数据流的干扰。Server 节点网卡为 100 Mbps 以太网卡，CPU 为 INTEL 双核 2.8 GHz，内存 4 GB，操作系统为 Linux2.6。Client 端为具有 ARM166MHz 的 CPU，64M 内存和 100M 以太网卡的嵌入式 IP 机顶盒。收发端的 TCP 和 UDP socket 的接收缓冲区大小均为 64 Kbytes。

在嵌入式机顶盒上使用 C 语言分别实现了 EMFTP、UDT 和 FTP 协议的接收端程序，并在 PC 上实现了 EMFTP 和 UDT 的发送端程序，FTP 的发送端使用的软件为 vsftpd。

2.1 传输性能

图 2 显示了分别使用 EMFTP、UDT、FTP 协议进行文件传输的速率曲线。下面从有效吞吐量和响应时间 2 个方面对 EMFTP、UDT 和 FTP 进行比较。

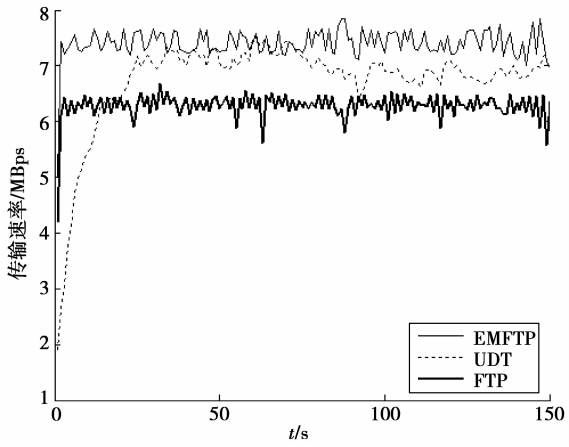

图 2 EMFTP、UDT、FTP 传输速率曲线比较

1)有效吞吐量

EMFTP 的速率控制机制允许发送方在传输过程中出现轻微丢包时不降低发送速率，丢失分组将在传输期结束时统一重传。这种机制在丢包率较低

的情况下，保证了传输速率的稳定性和快速性，能够提高有效吞吐量，从而获得更快的传输性能。从图 3 可以看出，EMFTP 的传输速率比 UDT 略高，FTP 的传输速率最低。

2）响应时间

响应时间是指从数据传输开始到达到稳定速度时所经历的时间。由于 EMFTP 的速度增量由探测出的带宽和当前发送速率决定，因此，在传输启动阶段，发送速率能够快速达到接近带宽容量的水平，然后进入速率控制的平稳期阶段，以一个较为平稳的速率传输数据。在带宽充足的情况下，EMFTP 和 FTP 都获得了较小的响应时间，而 UDT 响应时间较大，而且速率变化趋势较为缓慢。

表 1 描述了使用 3 种协议传输不同大小的文件所需的时间，表中时间是重复进行 20 次传输实验后的平均值。从表中可以看出，文件较小时，EMFTP 的性能提升并不明显，当文件比较大时，EMFTP 的传输时间比 UDT 缩短了 3%左右，比 FTP 缩短了 17%。这是由于文件较小时，EMFTP 的统一重传影响了整个文件的传输效率，但由于其协议开销较小，因此也能提高部分性能；当文件较大时，EMFTP 较为简单的应答机制就发挥了优势，获得了较为明显的性能提升。

表 1　文件传输消耗时间比较

文件大小(Bytes)	703M	1.5G	8.7G	14.3G
EMFTP/s	99	210	1 120	1 964
UDT/s	101	215	1 243	2 044
FTP/s	112	242	1 402	2 307

2.2　资源消耗水平

资源占用率是衡量嵌入式系统文件传输收发端能力匹配程度的重要指标。在实验环境下，发送方传输能力足够强，网络带宽足够大，因此，协议在嵌入式接收端的运行效率十分重要，需要避免 CPU 耗尽情况的发生。接收端资源占用率越低，说明协议对终端接收能力的适配性越强，能在较低的资源水平下运行。

衡量系统资源占用率的指标主要有 CPU 占用率和内存占用率这 2 个因素。图 3 对比了单条 EMFTP、UDT 和 FTP 传输流在嵌入式接收端的 CPU 占用率。EMFTP 在终端的 CPU 占用率最低，平均为 42.1%。FTP 其次，平均值为 84.4%。而 UDT 最高，平均 CPU 占用率达到 92.4%。

内存方面，EMFTP 的平均占用率为 8%，而 UDT 和 FTP 分别为 23%和 12%。

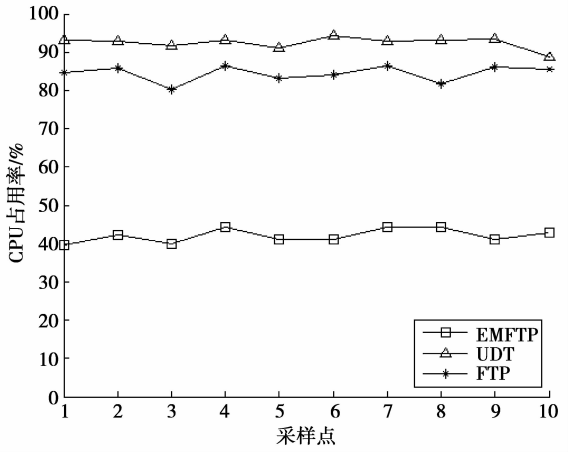

图 3　嵌入式接收端 CPU 占用率

实验表明，由于 EMFTP 采用了最为简单的接收方应答机制，最大限度降低了接收端的负担，因此在发送端传输能力和网络带宽都足够充分的情况下，其使用的 CPU 和内存资源远远少于 UDT 和 FTP，尤其是 CPU 占用率，不到后两者的一半，体现出了良好的收发端传输能力的匹配能力。

2.3　任务共存性

由于嵌入式 IP 机顶盒的主要任务是高清媒体播放等娱乐业务的高效开展，作为后台应用的文件传输必须降低系统资源的使用量，将更多的资源让给高优先级业务使用，不能影响高优先级业务的响应时间和运行效率。在协议与其他高优先级业务并存的情况下，业务共存性越高，业务受传输协议的影响就越小。

实验在文件传输过程中 50 s 时动态引入一个高清影片播放任务，在 100 s 时退出，对比 EMFTP、UDT 和 FTP 在有任务共存时的性能表现，及其对播放任务的影响。图 4 和图 5 分别对比了 3 种协议在 CPU 占用率和传输速率上的变化趋势。

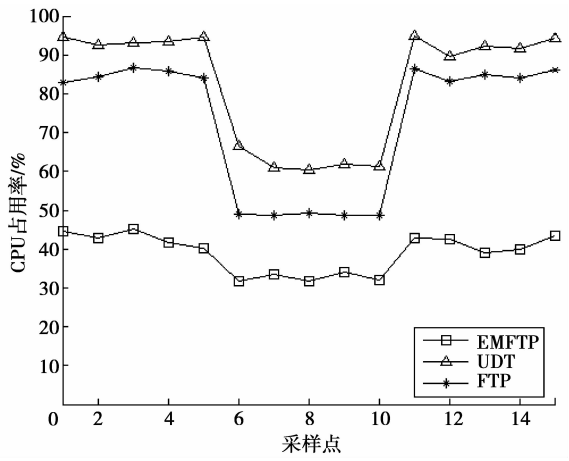

图 4　接收端负载动态变化时各协议的 CPU 占用率

由于 UDT 和 FTP 的 CPU 占用率过高，因此，当播放任务加入时，播放和文件传输之间会进行激烈的资源竞争，导致的结果是，播放任务无法获得足够的 CPU 资源，引起播放质量的下降。而 EMFTP 协议由于自身 CPU 占用率较低，与其他任务间的竞争较少，因此对播放任务的影响较小。表 2 给出了 3 种传输协议下播放质量和操作响应的对比，说明 EMFTP 对终端其他任务的影响最小。

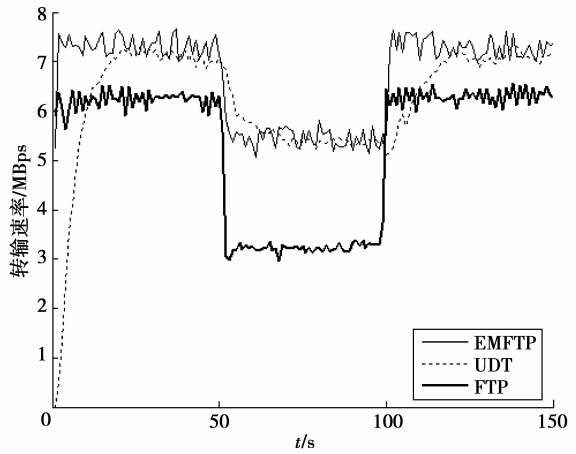

图 5 接收端负载动态变化时传输速率曲线

表 2 播放响应和质量对比

协议	播放质量	操作响应
EMFTP	播放流畅	及时
UDT	视频卡、顿现象明显	缓慢
FTP	偶尔出现视频卡、顿现象	及时

图 5 的传输曲线变化趋势说明，在播放任务接入后，由于系统资源产生了竞争，终端的网络数据接收处理能力进一步减弱，此时，3 种传输协议的有效吞吐量都有所降低。其中，EMFTP 和 FTP 能够很好的跟踪终端状态的变化，较快的达到稳定传输状态，而 UDT 传输速率的下降和上升都较慢，进一步验证了对 3 种协议响应时间的分析。

通过对比存在其他任务并行情况下 3 种协议的任务共存性，说明了 EMFTP 对终端接收能力具有很强的适配性，能更快的匹配收发端文件传输能力的差异。

2.4 公平性

公平性表示并发流共享链路带宽的公平度，常用的衡量准则是 Max-Min 公平准则[14]。如果链路中存在一条瓶颈路段，则所有并发的流应该平等的共享带宽。

为了评价数据流 a、b 在给定时间间隔 δ 内的公平性，采用公平度函数 $e_{\delta,a,b}(t)$，其定义如下[15]

$$e_{\delta,a,b}(t)=\min\left(\frac{r_{\delta,a}}{r_{\delta,b}},\frac{r_{\delta,b}}{r_{\delta,a}}\right);r_{\delta,a}>0,r_{\delta,b}>0,(6)$$

其中：$r_{\delta,a}$、$r_{\delta,b}$ 分别为 a、b 在 δ 内的平均传输速率；$e_{\delta,a,b}(t)$ 的值越接近 1 且变化平缓，称 2 种流越“公平”[16]。公平度量包括同类流和异类流间的公平度，分别讨论 EMFTP 流间的公平性和 EMFTP 流对 TCP 的友好性(即对 TCP 的公平性)。

2.4.1 EMFTP 流间的公平性

实验采用 3 个 Server 节点和 3 个 Client 节点，双方通过有效带宽为 100Mbps 的链路连接。这 3 对节点中，每隔 100 s 启动一个新的 EMFTP 流，然后再按照倒序依次关闭。图 6 描述了 EMFTP 的公平性。从图 6 中可以看到，当新的 EMFTP 流启动后，原先存在的 EMFTP 数据流会降低发送速率，与新的 EMFTP 流公平的共享带宽。

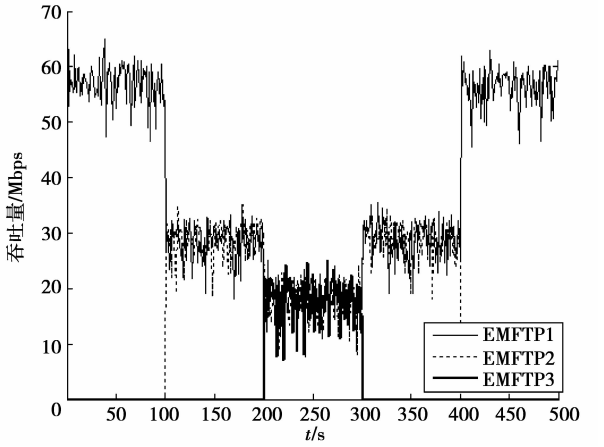

图 6 EMFTP 公平性

2.4.2 TCP 友好性

实验中首先启动一个 EMFTP 流作为背景传输流，在 100 s 时启动 TCP 传输流，图 7 显示，TCP 流启动后，EMFTP 流降低了数据传输的速率，让出一定的网络带宽给 TCP 流。

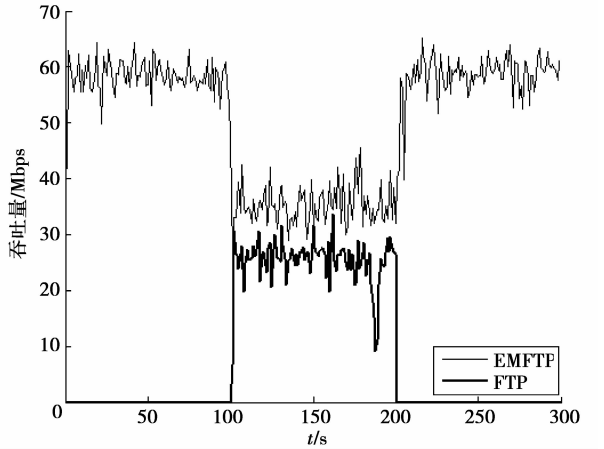

图 7 EMFTP 对 TCP 的友好性

文献[8]给出了 UDT 的 TCP 友好性曲线。虽然 EMFTP 能够让出部分带宽供 TCP 使用，但是与 UDT 相比，仍然具有一定的侵略性。EMFTP 通过牺牲部分友好性，以换取传输效率的提升。在下一步的工作中，将研究在保证性能的情况下，改善 EMFTP 的 TCP 友好性。

3　结　语

在网络带宽充足的条件下，较强的发送方发送能力和有限的嵌入式终端数据接收能力的差异，对终端提出了更高的资源需求。设计了一种用于匹配收发端数据传输能力的大型多媒体文件传输协议 EMFTP。该协议使用统一重传的差错恢复策略和基于带宽探测的 DAIMD 速率控制算法，简化了终端的应答机制，降低了终端资源消耗水平。通过实验验证了 EMFTP 在传输效率和资源消耗水平方面，均优于传统的文件传输协议 UDT 和 FTP。通过在文件传输过程中动态引入高清播放等高优先级任务，对比了 3 种传输协议对系统中同时运行的其他任务的影响，结果表明，在网络带宽足够大，发送端发送能力足够强的情况下，EMFTP 能够快速跟踪系统的负载情况，对接收端数据接收能力的适配性更好，对其他任务的影响最小，比 UDT 和 FTP 更适用于嵌入式系统的大型媒体文件传输。最后，实验分析出在公平性方面，EMFTP 对 TCP 仍具有一定的侵略性，这也是下一步工作需要改进之处。

参考文献：

[1] HA S, RHEE I, XU L. CUBIC: a new TCP-friendly high-speed TCP variant[J]. ACM SIGOPS Operating Systems Review, 2008, 42(5):64-74.

[2] KATABI D, HANDLEY M, ROHRS C. Congestion control for high bandwidth-delay product networks[C]//Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2002:89-102.

[3] XIA Y, SUBRAMANIAN L, STOICA I, et al. One more bit is enough[J]. IEEE/ACM Transactions on Networking (TON), 2008, 16(6):1281-1294.

[4] ZHANG Y, KANG S, LOGUINOV D. Delayed stability and performance of distributed congestion control[C]// Proceedings of the ACM SIGCOMM. Portland, Oregon. USA:[s. n.], 2004: 307-318.

[5] SOLLINS K, CHIAPPA N. The TFTP protocol[J]. Massachusetts Institute of Technology, IEN, 1980, 133(4): 25-34.

[6] HE E, LEIGH J, YU O, et al. Reliable Blast UDP: predictable high performance bulk data transfer[C]// IEEE Cluster Computing 2002. Chicago:[s. n.], 2002: 317-324.

[7] ANGLANO C, CANONICO M. A comparative evaluation of high-performance file transfer systems for data-intensive grid applications [C] // Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. VS: IEEE, 2004, 6:283-288.

[8] GU Y, GROSSMAN R. UDT: UDP-based data transfer for high-speed wide area networks [J]. Computer Networks, 2007, 51(7):1777-1799.

[9] DUNKELS A. Design and implementation of the lwIP TCP/IP stack[EB/OL]. Swedish Institute of Computer Science, 2001,2. [2010-04-28]. http://www.sics.se/~adam/lwip/doc/lwip.pdf.

[10] SHELBY Z, MAHONEN P, RIIHIJARVI J, et al. NanoIP: the zen of embedded networking[C]// IEEE International Conference on Communications (ICC2003). [s. l.]: Anchorage, 2003, 2:1218-1222.

[11] 张恒巍，万伟，王坤，等. 嵌入式可靠传输协议[J]. 计算机工程与设计，2010，31(2):327-329.
ZHANG HENG-WEI, WAN WEI, WANG KUN, et al. Embedded reliable transmission protocol [J]. Computer Engineering and Design, 2010, 31 (2): 327-329.

[12] GU Y, HONG X, GROSSMAN R. An analysis of AIMD algorithms with decreasing increases [C] // Gridnets 2004. First Workshop on Networks for Grid Applications. USA: Gridnets, 2004:89-101.

[13] ALLMAN M, PAXSON V. On estimating end-to-end network path properties[C]// ACM SIGCOMM '99, Cambridge: M A, 1999,9:28-31.

[14] NACE D, DOAN L, KLOPFENSTEIN O, et al. Max-min fairness in multi-commodity flows[J]. Computers and Operations Research, 2008, 35(2):557-573.

[15] FLOYD S, HANDLEY M, PADHYE J. Equation-based congestion control for unicast applications[C]// Proceeding of SIGCOMM' 2000. Stockholm, Sweden: ACM, 2000, 43-56.

[16] 朱晓亮，杜旭，杨宗凯. 一种适用于流媒体传输的无线 TCP 友好速率控制机制[J]. 小型微型计算机系统，2007,28(4):577-582。
ZHU XIAO-LIANG, DU XU, YANG ZONG-KAI. A wireless TCP-friendly rate control mechanism for streaming delivery[J]. Journal of Chinese Computer Systems, 2007, 28(4):577-582.

（编辑　侯　湘）