Transport Working Group R. Penno
Internet Draft Juniper Networks
Intended status: Informational J. Iyengar
Expires: May 2009 Franklin & Marshall College
November 3, 2008

TANA Practices and Recommendations
draft-penno-tana-app-practices-recommendation-01.txt

Status of this Memo

Copyright Notice

Abstract

Applications routinely open multiple TCP connections. For example, P2P applications maintain connections to a number of different peers while web browsers perform concurrent download from the same web server. Application designers pursue different goals when doing so:

P2P apps need to maintain a well-connected mesh in the swarm while web browsers mainly use multiple connections to parallelize requests that involve application latency on the web server side. But this practice also has impacts to the host and the network as a whole. For example, an application can obtain a larger fraction of the bottleneck than if it had used fewer connections. Although capacity is the most commonly considered bottleneck resource, middlebox state table entries are also an important resource for an end system communication.

This documents clarifies the current practices of application design and reasons behind them, and discusses the tradeoffs surrounding the use of many concurrent TCP connections to one destination and/or to different destinations. Other resource types may exist, and the guidelines are expected to comprehensively discuss them.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 Error! Reference source not found..

Table of Contents

1. Introduction

The use of P2P protocols by end users is widespread. These protocols are meant to exchange, replicate, stream or download files with little human intervention, trying at the same time to minimize the download time of the files requested by any single peer. This is done by opening several connections to multiple peers and downloading one or more chunks of the file from each one, selecting faster peers, amongst others.

If we assume that in any file transfer the bottleneck is on the uploading peer or server side, end users that utilize P2P clients in general download the file faster and consume more bandwidth within a specific timeframe than traditional client-server applications. P2P clients can overcome the server side bottleneck by opening multiple connections to different peers. Users of P2P applications also consume bandwidth throughout the whole day since even after a file is fully downloaded it will continue to be shared with others users increasing the upstream bandwidth.

We can see then that the advantages of P2P applications come from the fact that they open multiple TCP connection to different peers in order to download multiple pieces of a file in parallel, always look for faster peers.

But the use of multiple TCP connections by an application is not new. Web Browsers have being it for a decade. But these are usually short-lived connections as opposed to long-lived connections. A long-lived connection in this document should be interpreted as strictly defined, i.e., a TCP connection that is simply in the established state, but not necessarily continuously transferring data. In the case of P2P protocols, e.g. Bittorrent, at any point in time a fraction of these connections is actually sending or receiving data, the others are idle or exchange occasional control information.

With the popularity of P2P applications, which maintain hundreds of long-lived TCP connections to multiple hosts, the issue applications making use of multiple TCP connections has been gaining attention.

This documents clarifies the current practices of application design and reasons behind them, and discusses the tradeoffs surrounding the use of many concurrent TCP connections to one destination and/or to different destinations. Other resource types may exist, and the guidelines are expected to comprehensively discuss them.

2. Terminology

Bandwidth: A measure of the amount of data that can be transferred per second. So, if a 1Gb file were transferred within one second the bandwidth consumption during the transfer would be 1Gb/s. If it were transferred within a day, it would be approximately 0.0002Gb/s.

Volume: The total number of bytes transferred during a long time period. In both examples above the volume within a day would have been 1Gb.

Capacity: The maximum bandwidth a link can sustain continuously.

Long-lived connection: A TCP connection that is in the established state but not necessarily continuously transferring data.

3. Multiple TCP Connections Advantages

There are good reasons for an application to use multiple TCP connections. P2P apps need to maintain a well-connected mesh in the swarm while web browsers mainly use multiple connections to parallelize requests that involve application latency on the web server side

But from a P2P standpoint multiple TCP connections are at the heart of its functionality. Multiple connections allow for multiple simultaneous downloads, which improve reliability and speed. Multiple connections also allow more effective discovery of new peers, and effective peer-to-peer communication, which allows exchange of information such as which pieces of a file a client has and is available.

4. Multiple TCP connections Resources

Every connected application on the Internet competes for resources. This is not specific to applications that open multiple TCP connections. The use of multiple TCP connections just amplifies the issue. In the following sections we discuss these resources and how they are amplified by an application opening multiple connections.

5. Memory Space

Each TCP connection needs a TCP control block (TCB) or equivalent to keep state about its connection. In operating systems where the TCP stack is part of the kernel, this would come from the kernel memory space, otherwise from userland memory.

But irrespective from where the memory comes from a TCP control block requires a significant amount of memory. This is significant issue for devices that terminate TCP connections from multiple end hosts to provide functions such as Load-Balancing, Gateway and Tunneling.

Some proposals have been put forward to reduce the amount of memory occupied by each TCP control block [RFC2140], but the issue remains significant and is amplified by applications that use multiple TCP connections.

6. Link Bandwidth

The bottlenecks for these n connections could be shared or separate. If separate, there's no specific bottleneck where the connections are hogging bandwidth. But from a network resource point of view, the application download still gets multiple shares.

If some/all bottlenecks are shared, then two possibilities exist for shared bottleneck

- bottleneck is a last-hop link (user traffic dominates link), OR

- bottleneck is in-network wide-area link (background traffic dominates link)

If bottleneck is last-hop, then n transport connections compete with each other and share link bandwidth.

Although these connections might impact delay-sensitive traffic and increase delay, in the last hop it only affects end end-user, which is in control of which applications run on its host. In this case the user has the option of manually choosing when to run such application, configuring the end host, amongst others. Alternatively, or in conjunction with the above, the application can be enhanced to use Diffserv and new delay sensitive congestion mechanisms.

If shared bottleneck is in-network, then application gets unfair share of bottleneck bandwidth. This impacts flows belonging to other users in general, and most importantly delay-sensitive traffic.

7. Middleboxes

Middleboxes re defined as any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host [RFC3234]. Middleboxes can be stand-alone or integrated in another device such as a router or modem.

The functions that are relevant to this discussion are those that require the middlebox to keep per session state, sometimes referred as transformation services. Some of these functions are, for example, NAT, Intrusion Detection and Load-Balancing.

It is easy to see that the more sessions a host initiates, the more state the middlebox will have to keep. The relationship is at least 1:1 but due asymmetric traffic, routing changes and others, this can be 1:N.

Although application traffic from most broadband subscribers today go through at least one middlebox (integrated into the broadband modem), it can traverse other middleboxes that reside within the ISP's network or close the destination. These middleboxes aggregate traffic from multiple subscribers and state tables within these devices can become a premium.

8. Recommendations

8.1. Diffserv

8.2. AQM

9. Security Considerations

None at this time

10. IANA Considerations

None at this time

11. Conclusions

TBD

12. Acknowledgments

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

13.2. Informative References

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC2140] J. Touch, TCP Control Block Interdependence, RFC 2140 (1997)

Author's Addresses

Reinaldo Penno
Juniper Networks
1194 N Mathilda Aveue
Sunnyvale, CA

Email: rpenno@juniper.net

Jana Iyengar
Franklin & Marshall College

Email: jiyengar@fandm.edu

Intellectual Property Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.