METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

RELATED APPLICATIONS

~~[0001~~

[0001] This application is a continuation-in-part of, and claims priority to U.S. Patent Application No. 14/667,642, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SELECTING A RESOURCE BASED ON A MEASURE OF A PROCESSING COST," filed on 03-24-2015 which, in turn, is a continuation-in-part of and claims priority to U.S. Patent Application No. 13/477,402, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION," filed 05-22-2012 which is a continuation of and claims priority to U.S. Patent Application No. 12/714,454, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION," filed 02- 27-2010.

[0002] U.S. Patent Application No. 12/714,454, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION," filed 02-27-2010 is incorporated herein by reference in its entirety for all purposes.

[0003] This application is related to the following commonly owned U.S. Patent Applications, the entire disclosure of ~~each being~~which is incorporated by reference herein in its entirety for all purposes: Application No. 12/714,063 (Docket No 0110) filed on 2010/02/26, entitled "Methods, Systems, and Program Products for Detecting an Idle TCP Connection".

BACKGROUND

[~~0002~~0004] Various implementations of the transmission control protocol (TCP) in network nodes support a number of options that are not negotiated or even communicated between or among any of the nodes. Some of these options are included in the specification of the TCP while others are not. For example, the TCP keep-alive option is supported by a number of implementations of the TCP. It is not, however, part of the TCP specification as described in "Request for Comments" (RFC) document RFC 793 edited by John Postel, titled "Transmission Control Protocol, DARPA Internet Program Internet Protocol Specification" (September 1981), which is incorporated here in its entirety by reference. One, both, or neither node including an endpoint in a TCP connection may support a keep-alive option for the connection. Each node supports or does not support keep-alive for a TCP connection based on each node's requirements without consideration for the other node in the TCP connection.

[~~0003~~0005] With respect to the keep-alive option, some argue that it is unnecessary and that it can waste network bandwidth. Some of these critics point out that a keep-alive packet can bring down a TCP connection. Further, since nodes including endpoints in a TCP connection do not cooperate in supporting the keep-alive option, the nodes may operate in opposition to one another and/or may waste resources by duplicating function, according to critics of the keep-alive option.

[~~0004~~0006] Proponents of the keep-alive option claim there is a benefit to detecting a dead peer/partner endpoint sooner. A node providing TCP keep-alive can also indirectly detect when a network is so congested that two nodes with endpoints in a TCP connection are effectively disconnected. Proponents argue that keep-alive can keep an inactive TCP connection open. For example, some network nodes such as firewalls are configured to close TCP connections determined to be idle or inactive in order to recover resources. Keep-alive can prevent this. This is good from the perspective of the node sending keep-alive packets, but the keep-alive packets might cause the firewall to waste resources and possibly block or terminate TCP connections with other nodes.

[~~0005~~0007] TCP keep-alive and the debate of its benefits and faults have been around for decades. To date no mechanism to allow two TCP connection endpoints to cooperate in supporting the keep-alive option has been proposed or implemented. The broader issue of enabling cooperation and negotiation between nodes in a TCP connection in detecting and managing idle, underactive, and/or dead TCP connections remains unaddressed.

[~~0006~~0008] Accordingly, there exists a need for methods, systems, and computer program products for sharing information for detecting an idle TCP connection.

SUMMARY

[~~0007~~0009] The following presents a simplified summary of the disclosure in order to provide a basic understanding to the reader. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[~~0008] Methods~~0010] An apparatus is provided comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for: receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established; detecting an idle time period parameter field in the TCP-variant packet; identifying metadata in the idle time period parameter field for an idle time period that is detectable by the first node and, during which, no packet is communicated in the TCP-variant connection to keep the TCP- variant connection active; and modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.

[0011] Another apparatus is provided comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for: receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant

connection active; generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and sending, from a first node to a second node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the second node, for use by the second node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.

[0012] Yet another apparatus is provided comprising: a non-transitory memory storing a network application; and one or more processors in communication with the non- transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a non- transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the apparatus, when operating in accordance with the non-TCP protocol, configured to: receive, by a second node from a first node, a non-TCP packet during a setup of a non- TCP connection; identify metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period that is detectable by the first node, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation; and determine, based on the metadata, a timeout attribute associated with the non-TCP connection; wherein the apparatus, when operating in accordance with the TCP protocol, is configured to perform a three-

way TCP handshake for establishing a TCP connection that is different than the non- TCP connection.

[0013] Still yet another apparatus is provided comprising: a non-transitory memory storing a network application; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the apparatus, when operating in accordance with the non-TCP protocol, configured to: receive idle information for use in detecting an idle time period that results in a non-TCP connection being subject to deactivation; generate, based on the idle information, a non-TCP packet including an idle time period parameter field identifying metadata that is specified in a number of seconds or minutes; and send, from a first node to a second node and for establishing the non-TCP connection, the non-TCP packet to provide the metadata to the second node, for use by the second node in determining a timeout attribute associated with the non-TCP connection; wherein the apparatus, when operating in accordance with the TCP protocol, is configured to perform a three-way TCP handshake for establishing a TCP connection that is separate from the non-TCP connection.
[0014] Other methods and systems are also described for sharing information for detecting an idle TCP connection. In one aspect, a method includes receiving, by a

second node from a first node, a first transmission control protocol (TCP) packet in a TCP connection. The method further includes detecting a first idle time period header, in the first packet, identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by the second node is received by the first node. The method still further includes modifying, based on the metadata, by the second node a timeout attribute associated with the TCP connection.

[~~0009~~0015] Further, a system for sharing information for detecting an idle TCP connection is described. The system includes an execution environment including an instruction processing unit configured to process an instruction included in at least one of a net in- port component, an idle time period option handler component, and an option attribute handler component. The system includes the net in-port component configured for receiving, by a second node from a first node, a first transmission control protocol (TCP) packet in a TCP connection. The system further includes the idle time period option handler component configured for detecting a first idle time period header, in the first packet, identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by the second node is received by the first node. The system still further includes the option attribute handler component configured for modifying, based on the metadata, by the second node a timeout attribute associated with the TCP connection

[~~0010~~0016] In another aspect, a method for sharing information for detecting an idle TCP connection is described that includes receiving, by a first node, first idle information for detecting a first idle time period during which no TCP packet including data in a first data stream sent in the TCP connection by a second node is received by the first node. The method further includes generating a TCP packet including a first idle time period header identifying metadata for the first idle time period based on the first idle information. The method still further includes sending the TCP packet in the TCP connection to the second node to provide the metadata for the first idle time period to the second node. The method also includes detecting the first idle time period based on the first idle information. The method additionally includes deactivating the TCP connection in response to detecting the first idle time period.

[~~0011~~0017] Still further, a system for sharing information for detecting an idle TCP connection is described. The system includes an execution environment including an instruction processing unit configured to process an instruction included in at least one of an idle time period policy component, a packet generator component, a net out-port component, an idle time period monitor component, and a connection state component. The system includes the idle time period policy component configured for receiving, by a first node, first idle information for detecting a first idle time period during which no TCP packet including data in a first data stream sent in the TCP connection by a second node is received by the first node. The system includes the packet generator component configured for generating a TCP packet including a first idle time period

header identifying metadata for the first idle time period based on the first idle information. The system still further includes the net out-port component configured for sending the TCP packet in the TCP connection to the second node to provide the metadata for the first idle time period to the second node. The system includes the idle time period monitor component configured for detecting the first idle time period based on the first idle information. The system includes the connection state component configured for deactivating the TCP connection in response to detecting the first idle time period.

BRIEF DESCRIPTION OF THE DRAWINGS

[~~0012~~0018] Objects and advantages of the present invention will become apparent to those skilled in the art upon reading this description in conjunction with the accompanying drawings, in which like reference numerals have been used to designate like or analogous elements, and in which:

[~~0013~~0019] Fig. 1 is a block diagram illustrating an exemplary hardware device included in and/or otherwise providing an execution environment in which the subject matter may be implemented;

[~~0014~~0020] Fig. 2 is a flow diagram illustrating a method for sharing information for detecting an idle TCP connection according to an aspect of the subject matter described herein;

[~~0015~~0021] Fig. 3 is a flow diagram illustrating another method for sharing information for detecting an idle TCP connection according to another aspect of the subject matter described herein;

[~~0016~~0022] Fig. 4 is a block a diagram illustrating an arrangement of components for sharing information for detecting an idle TCP connection according to a further aspect of the subject matter described herein;

[~~0017~~0023] Fig. 5 is a block diagram illustrating an arrangement of components for sharing information for detecting an idle TCP connection according to still another aspect of the subject matter described herein;

[~~0018~~0024] Fig. 6 is a network diagram illustrating an exemplary system for sharing information for detecting an idle TCP connection according to an aspect of the subject matter described herein;

[~~0019~~0025] Fig. 7 is a message flow diagram illustrating an exemplary data and execution flow for sharing information for detecting an idle TCP connection according to an aspect of the subject matter described herein; and

[~~0020~~0026] Fig. 8 is a diagram illustrating a structure for a packet transmitted via a network according to an aspect of the subject matter described herein.

DETAILED DESCRIPTION

[~~0021~~0027] An exemplary device included in an execution environment that may be configured according to the subject matter is illustrated in Fig. 1. An execution environment includes an arrangement of hardware and, optionally, software that may be further configured to include an arrangement of components for performing a method of the subject matter described herein.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**