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

RELATED APPLICATIONS

[0001] This application is a continuation of, and claims priority to U.S. Patent Application No. 16/914,267 entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION, " filed on 06-26-2020, which in turn is a continuation of, and claims priority to U.S. Patent Application No. 16/368,811 entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION," filed on 03-28-2019, which in turn is a continuation of, and claims priority to U.S. Patent Application No. 16/040,522, now US Patent Number 10,375,215, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION, " filed on 07-19-2018, which in turn, is a continuation of, and claims priority to U.S. Patent Application No. 15/915,047, now U.S. Patent No. 10,075,564, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION, " filed on 03-07-2018 which, in turn, is a continuation of, and claims priority to U.S. Patent Application No. 15/694,802, now U.S. Patent No. 9,923,995, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION, " filed on 09-03-2017

which, in turn, 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, now U.S. Patent No. 8,219,606, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP

[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

CONNECTION, " filed 02-27-2010-27-2010.

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 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

DOCKET

[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.

[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.

[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.

[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.

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

SUMMARY

DOCKET

[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.

[0010] An apparatusA computer-implemented method is provided, comprising: causing access to be provided to a server computer including: a non-transitory memory storing instructions; a network application, 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 fromnetwork application to operate in accordance with a first node, protocol including a transmission control protocol (TCP) -variant packet in advance of : causing a TCP-variant connection beingto be established; with a client computer, by: communicating a segment including at least one first synchronize bit, communicating a first acknowledgement of the segment, and at least one second synchronize bit, and communicating a second acknowledgement; causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer; causing the server computer to permit second data, from the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and causing access to be provided, to the client computer, to code that causes the client computer to operate in accordance with a second protocol that is separate from the TCP, in order to establish a second protocol connection with another server computer, by: receiving a packet, detecting an idle time period parameter field in the TCP variant packet; identifying

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; where, after the idle time period is detected, the second protocol connection is deemed inactive, and creating or modifying, by the second nodeclient computer and based on the metadata, a timeout attribute associated with the <u>TCP variant</u> second protocol connection.

metadata in the idle time period

DOCKET

[0011] Another apparatus computer-implemented method is provided comprising: providing access to a server computer including: a non-transitory memory storing instructions; a network application, 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 keepnetwork application to operate in accordance with a first protocol including a transmission control protocol (TCP); causing a TCP connection to be established with a client computer, by communicating a segment including at least one first synchronize bit; communicating a first acknowledgement of the segment, and at least one second synchronize bit; and communicating a second acknowledgement; causing first data to be communicated from the server computer to the client computer utilizing the TCP-variant connection active; in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer; causing the server computer to permit second data, from the user of the client

computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and providing access to code that results in the client computer operating in accordance with a second protocol that

A L A R M Find authenticated court documents without watermarks at <u>docketalarm.com</u>.

is separate from the TCP, in order to establish a second protocol connection with another

server computer, by: identifying idle information for detecting an idle time period, after which, the second protocol connection is subject to deactivation, generating a TCP-variantsecond protocol 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 the client computer to the another server computer, the second node, the TCP-variantprotocol packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the second nodeanother server computer, for use by the second nodeanother server computer in creating or modifying, based on the metadata, a timeout attribute associated with the TCP-variantsecond protocol connection.

[0012] Yet another apparatus computer-implemented method is provided comprising: providing access to a server computer including: a non-transitory memory storing a network application; instructions, and one or more processors in communication with the non- transitory memory, wherein the one or more processors execute the network application instructions such that thea network application is configured to operate operates in accordance with a non first protocol including a transmission control protocol (TCP) protocol, the server computer, when operating in accordance with the first protocol to set up a TCP connection with a client computer, configured to: communicate a segment including at least one first synchronize bit, communicate a first acknowledgement of the segment, and at least one second synchronize bit, and communicate a second acknowledgement; causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer; causing the server computer to permit second data, of the user of the client

computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and providing access to code that causes the client computer to operate in accordance with a second protocol that is different from the TCP and 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 in order to setup a second protocol connection with another server computer, by: receiving, by the client computer from the another server computer, a non-TCP packet during a setup of a non- TCP connection; identify, identifying 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; during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive, and determining, by the client computer and determine, based on the metadata, a timeout attribute associated with the non-TCP connection; wherein the apparatus, when operating in accordance with the TCPsecond protocol, is configured to perform a three-

A L A R M Find authenticated court documents without watermarks at <u>docketalarm.com</u>.

DOCKET

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

[0013] Still yet another apparatus_computer-implemented method is provided comprising: providing access to a server computer including: a non-transitory memory storing a network application; instructions, and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application_instructions such that thea network application is configured to operateoperates in accordance with a non-first protocol including a transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the apparatus_server computer, when operating in accordance with the non-TCP_first protocol to set up a TCP connection with a client

computer, configured to: communicate a segment including at least one first synchronize bit, communicate a first acknowledgement of the segment, and at least one second synchronize bit, and communicate a second acknowledgement; causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer; causing the server computer to permit second data, of the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and providing access to code that results in the client computer operating in accordance with a second protocol, that is different from the TCP and operates above the IP layer and below the HTTP application layer, in order to setup a second protocol connection with another server computer, and to: receive idle information for use in detecting an idle time period that during which no signal is communicated that meets each of the following criteria: a) communicated in the second protocol connection, and b) results in a non TCPthe second protocol connection being subject to deactivation; at least partially kept alive, generate, based on the idle information, a non-TCPsecond protocol 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 the client computer to another server computer and for establishingduring the non-TCPset up of the second protocol connection, the non-TCPsecond protocol packet to provide the metadata to the second node another server computer, for use by the second nodeanother server computer 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-TCPsecond protocol 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

DOCKET

LARM Find authenticated court documents without watermarks at <u>docketalarm.com</u>.

DOCKET



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.

