United States Patent Application Publication	20250254063
Kind Code	A1
Publication Date	August 07, 2025
Inventor(s)	RUBENSTEIN; Joseph E. et al.

SYSTEM AND METHOD FOR A GLOBAL VIRTUAL NETWORK

Abstract

Systems and methods for connecting devices via a virtual global network are disclosed. In one embodiment the network system may comprise a first device in communication with a first endpoint device and a second device in communication with a second endpoint device. The first and second devices may be connected with a communication path. The communication path may comprise one or more intermediate tunnels connecting each endpoint device to one or more intermediate access point servers and one or more control servers.

Inventors: **RUBENSTEIN; Joseph E. (Beijing, CN), KNUTSEN; Jørn Allan Dose (Oslo, NO), SAINT-MARTIN; Thibaud August Bernard Jean (Aubignan, FR), ORÉ; Carlos Eduardo (Saint-Herblain, FR), BROUSSARD; Fred (Indianapolis, IN)**

Applicant: **UMBRA Technologies Ltd.** (Hong Kong, CN)

Family ID: **56544309**

Assignee: **UMBRA Technologies Ltd. (Hong Kong, HK)**

Appl. No.: **19/188571**

Filed: **April 24, 2025**

Related U.S. Application Data

parent US continuation 18390894 20231220 parent-grant-document US 12289183 child US 19188571
parent US continuation 17589753 20220131 parent-grant-document US 11881964 child US 18390894
parent US continuation 16815864 20200311 parent-grant-document US 11240064 child US 17589753
parent US continuation 15546247 20170725 parent-grant-document US 10630505 US continuation PCT/US16/15278 20160128 child US 16815864
us-provisional-application US 62266060 20151211

us-provisional-application US 62174394 20150611
us-provisional-application US 62151174 20150422
us-provisional-application US 62144293 20150407
us-provisional-application US 62108987 20150128

Publication Classification

Int. Cl.: **H04L12/46** (20060101); **H04L9/40** (20220101); **H04L45/12** (20220101); **H04L45/64** (20220101); **H04L61/4511** (20220101)

U.S. Cl.:

CPC **H04L12/4633** (20130101); **H04L45/12** (20130101); **H04L45/64** (20130101); **H04L61/4511** (20220501); **H04L63/02** (20130101);

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATIONS [0001] This application is a continuation of U.S. application Ser. No. 18/390,894, filed Dec. 20, 2023, which is a continuation of U.S. application Ser. No. 17/589,753, filed Jan. 31, 2022, now U.S. Pat. No. 11,881,964, which is a continuation of U.S. application Ser. No. 16/815,864, filed on Mar. 11, 2020, now U.S. Pat. No. 11,240,064, which is a continuation of U.S. application Ser. No. 15/546,247, filed on Jul. 25, 2017, now U.S. Pat. No. 10,630,505, which is a U.S. National Stage Application under 35 U.S.C. § 371 of International Patent Application No. PCT/US16/15278, filed on Jan. 28, 2016, which claims priority to U.S. Provisional Application No. 62/266,060, filed Dec. 11, 2015; U.S. Provisional Application No. 62/174,394, filed Jun. 11, 2015; U.S. Provisional Application No. 62/151,174, filed Apr. 22, 2015; U.S. Provisional Application No. 62/144,293 filed Apr. 7, 2015; and U.S. Provisional Application No. 62/108,987, filed Jan. 28, 2015; and International Application No. PCT/US16/15278 is a continuation of International Application No. PCT/IB16/00110, filed on Jan. 5, 2016 and International Application No. PCT/US15/64242 filed Dec. 7, 2015, all of which are incorporated herein by reference in their entireties. U.S. Provisional Application No. 62/089,113 filed on Dec. 8, 2014, and U.S. Provisional Application No. U.S. 62/100,406 filed on Jan. 6, 2015, are incorporated herein by reference.

FIELD OF THE DISCLOSURE

[0002] The present disclosure relates generally to networks, and more particularly, to the configuration and operation of a global virtual network (GVN).

BACKGROUND OF THE DISCLOSURE

[0003] While last mile connectivity has vastly improved in recent years there still exist problems with long distance connectivity and throughput due to issues related to distance, protocol limitations, peering, interference, and other problems and threats. A GVN offers secure network optimization services to clients over the top of their standard internet connection.

[0004] This is an overview of the constituent parts of a GVN as well as a description of related technologies which can serve as GVN elements. GVN elements may operate independently or within the ecosystem of a GVN such as utilizing the GVN framework for their own purposes, or can be deployed to enhance the performance and efficiency of a GVN.

[0005] This overview also describes how other technologies can benefit from a GVN either as a stand-alone deployment using some or all components of a GVN, or which could be rapidly deployed as an independent mechanism on top of an existing GVN, utilizing its benefits.

[0006] Human beings are able to perceive delays of 200 ms or more as this is typically the average human reaction time to an event. If latency is too high, online systems such as thin-clients to cloud-based servers, customer relationship management (CRM), enterprise resource planning (ERP) and other systems will perform poorly and may even cease functioning due to timeouts. High latency combined with high packet loss can make a connection unusable. Even if data gets through, at a certain point too much slowness results in a poor user experience (UX) and in those instances the result can be refusal by users to accept those conditions in effect rendering poorly delivered services as useless.

[0007] To address some of these issues, various technologies have been developed. One such technology is WAN optimization, typically involving a hardware (HW) device at the edge of a local area network (LAN) which builds a tunnel to another WAN optimization HW device at the edge of another LAN, forming a wide area network (WAN) between them. This technology assumes a stable connection through which the two devices connect to each other. A WAN optimizer strives to compress and secure the data flow often resulting in a speed gain. The commercial driver for the adoption of WAN optimization is to save on the volume of data sent in an effort to reduce the cost of data transmission. Disadvantages of this are that it is often point-to-point and can struggle when the connection between the two devices is not good as there is little to no control over the path of the flow of traffic through the Internet between them. To address this, users of WAN optimizers often opt to run their WAN over an MPLS or DDN line or other dedicated circuit resulting in an added expense and again usually entailing a rigid, fixed point-to-point connection.

[0008] In the marketplace at the time of writing of this patent, a group of venders focus on selling hardware but not connection services on the internet between their hardware devices. Another group of vendors are service providers who may provide a simple end point device or software which can be installed by their customers onto their own devices to connect to their service provider's cloud servers, as a link to the services that they provide as a bundle, but their main focus is on the provision of the services.

[0009] Direct links such as MPLS, DDN, Dedicated Circuits or other types of fixed point-to-point connection offer quality of connection and Quality of Service (QoS) guarantees. They are expensive and often take a significantly long time to install due to the need to physically draw lines from a POP at each side of the connection. The point-to-point topology works well when connecting from within one LAN to the resources of another LAN via this directly connected WAN. However, when the gateway (GW) to the general Internet is located at the LAN of one end, say at the corporate headquarters, then traffic from the remote LAN of a subsidiary country may be routed to the Internet through the GW. A slowdown occurs as traffic flows through the internet back to servers in the same country as the subsidiary. Traffic must then go from the LAN through the WAN to the LAN where the GW is located and then through the Internet back to a server in the origin country, then back through the internet to the GW, and then back down the dedicated line to the client device within the LAN. In essence doubling or tripling (or worse) the global transit time of what should take a small fraction of global latency to access this nearby site. To overcome this, alternative connectivity of another internet line with appropriate configuration changes and added devices can offer local traffic to the internet, at each end of such a system.

[0010] Another option for creating WAN links from one LAN to another LAN involve the building of tunnels such as IPSec or other protocol tunnels between two routers, firewalls, or equivalent edge devices. These are usually encrypted and can offer compression and other logic to try to improve connectivity. There is little to no control over the routes between the two points as they rely on the policy of various middle players on the internet who carry their traffic over their network(s) and peer to other carriers and or network operators. Firewalls and routers, switches and other devices from a number of equipment vendors usually have tunneling options built into their firmware.

[0011] A software (SW) based virtual private network (VPN) offers privacy via a tunnel between a client device and a VPN server. These have an advantage of encryption and in some cases also compression. But here again there is little to no control over how traffic flows between VPN client and VPN server as well as between the VPN server and host server, host client or other devices at destination. These are often point-to-point connections that require client software to be installed per device using the VPN and some technical proficiency to maintain the connection for each device. If a VPN server egress point is in close proximity via quality communication path to destination host server or host client then performance will be good. If not, then there will be noticeable drags on performance and dissatisfaction from a usability perspective. It is often a requirement for a VPN user to have to disconnect from one VPN server and reconnect to another VPN server to have quality or local access to content from one region versus the content from another region.

[0012] A Global Virtual Network (GVN) is a type of computer network on top of the internet providing global secure network optimization services utilizing a mesh of devices distributed around the world securely linked to each other by advanced tunnels, collaborating and communicating via Application Program Interface (API), Database (DB) replication, and other methods. Traffic routing in the GVN is always via best communication path governed by Advanced Smart Routing (ASR) powered by automated systems which combine builders, managers, testers, algorithmic analysis and other methodologies to adapt to changing conditions and learning over time to configure and reconfigure the system.

[0013] The GVN offers a service to provide secure, reliable, fast, stable, precise and focused concurrent connectivity over the top of one or more regular Internet connections. These benefits are achieved through compression of data flow transiting multiple connections of wrapped, disguised and encrypted tunnels between the EPD and access point servers (SRV_AP) in close proximity to the EPD. The quality of connection between EPD and SRV_AP's is constantly being monitored.

[0014] A GVN is a combination of a hardware (HW) End Point Device (EPD) with installed software (SW), databases (D B) and other automated modules of the GVN system such as Neutral Application Programming Interface Mechanism (NAPIM), back channel manager, tunnel manager, and more features which connect the EPD to distributed infrastructure devices such as access point server (SRV_AP) and central server (SRV_CNTRL) within the GVN.

[0015] Algorithms continually analyze current network state while taking into account trailing trends plus long term historical performance to determine best route for traffic to take and which is the best SRV_AP or series of SRV_AP servers to push traffic through to. Configuration, communication path and other changes are made automatically and on the fly with minimal or no user interaction or intervention required.

[0016] Advanced Smart Routing in an EPD and in an SRV_AP ensure that traffic flows via the most ideal path from origin to destination through an as simple as possible “Third Layer” of the GVN. This third layer is seen by client devices connected to the GVN as a normal internet path but with a lower number of hops, better security and in most cases lower latency than traffic flowing through the regular internet to the same destination. Logic and automation operate at the “second layer” of the GVN where the software of the GVN automatically monitors and controls the underlying routing and construct of virtual interfaces (VIF), multiple tunnels and binding of communication paths. The third and second layers of the GVN exist on top of the operational “first layer” of the GVN which interacts with the devices of the underlying Internet network.

SUMMARY OF THE DISCLOSURE

[0017] Systems and methods for connecting devices via a virtual global network are disclosed. The network system may comprise a first device in communication with a first endpoint device. The network system may comprise a second device in communication with a second endpoint device. The first and second devices may be connected with a communication path. The communication path may comprise one or more intermediate tunnels connecting each endpoint device to one or more intermediate access point servers and one or more control servers.

[0018] In accordance with other aspects of this embodiment, at least one of the first end point device and the intermediate access point servers is configured to perform a Domain Name System (DNS) lookup to locate the second device.

[0019] In accordance with other aspects of this embodiment, at least one of the first end point device and the intermediate access point servers is configured to perform a Domain Name System (DNS) lookup from a cache to locate the second device.

[0020] In accordance with other aspects of this embodiment, at least one of the intermediate access point servers is configured to cache content.

[0021] In accordance with other aspects of this embodiment, at least one of the end point devices and the intermediate access point servers is configured to perform smart routing based on a global virtual network.

[0022] In accordance with other aspects of this embodiment, the smart routing is based on at least one of best bandwidth, lowest latency, fewest hops, and no packet loss.

[0023] In accordance with other aspects of this embodiment, the smart routing is based on at least one of real-time statistics and historical statistics.

[0024] In accordance with other aspects of this embodiment, at least one of the end point devices and the intermediate access point servers is configured to perform firewall services.

[0025] In accordance with other aspects of this embodiment, the firewall services are between the first device and the intermediate access point servers.

[0026] In accordance with other aspects of this embodiment, the firewall services are between the first device and the intermediate access point servers and second endpoint device.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] In order to facilitate a fuller understanding of the present disclosure, reference is now made to the accompanying drawings, in which like elements are referenced with like numerals or references. These drawings should not be construed as limiting the present disclosure, but are intended to be illustrative only.

[0028] FIG. **1** shows a block diagram of technology used by and enabled by a global virtual network (“GVN”).

[0029] FIG. **2** shows a high-level block diagram of the Internet.

[0030] FIG. **3** is a block diagram showing the resolution of Universal Resource Locator (URLs) to numeric Internet Protocol (IP) addresses via the Domain Name System (DNS).

[0031] FIG. **4** is a simplified illustration showing the upstream and downstream paths data takes from a host client device (C##) to another host client or host server device (S ##).

[0032] FIG. **5** is a simplified illustration showing the boundary switches in the paths data takes from a host client device (C##) to another host client or host server device (S ##).

[0033] FIG. **6** illustrates some example threats and problems which exist on the Internet.

[0034] FIG. **7** illustrates Content Delivery Network (CDN) resolution and delivery of regionally specific content.

[0035] FIG. **8** illustrates the operation of a proxy server.

[0036] FIG. **9** illustrates a point-to-point tunnel built between two gateway devices.

[0037] FIG. **10** shows the relationship of security features between device scope, system-wide scope, communications scope, and device collaboration.

[0038] FIG. **11** illustrates the information flow between devices of a Global Virtual Network.

[0039] FIG. **12** describes the stack for supporting the automation of some devices in a GVN.

[0040] FIG. **13** illustrates that GVN topology, including a backbone segment over internet or dark fiber.

[0041] FIG. **14** illustrates a distributed firewall (FW) in the cloud enabled by a GVN.
[0042] FIG. **15** illustrates a multi-perimeter firewall (MPFW) in the cloud empowered by a global virtual network.
[0043] FIG. **16** illustrates a logical view of the software architecture of three types of network devices working together as a part of a Global Virtual Network (GVN).
[0044] FIG. **17** illustrates a GVN using hub and spoke topology with a backbone and octagon routing.
[0045] FIG. **18** illustrates the backbone connections between some GVN Global Nodes and their corresponding service areas in North America, Europe and Asia.
[0046] FIG. **19** illustrates the connectivity between various devices within a GVN.
[0047] FIG. **20** illustrates how GVN modules and devices interact.
[0048] FIG. **21** illustrates additional details of how GVN modules and devices interact.
[0049] FIG. **22** illustrates how GVN modules and devices interact with other devices on the Internet.
[0050] FIG. **23** illustrates multiple tunnel connectivity between End Point Devices (EPD) and Access Point Servers (SRV_AP).
[0051] FIG. **24** is a simplified example diagram of how the internet works today taking into account hop count or time to live (TTL) as well as path taken due to peering relationships and related routing policies.
[0052] FIG. **25** illustrates the strategic positioning of infrastructure to enhance performance.
[0053] FIG. **26** illustrates how the GVN can incorporate technologies such as Network Slingshot.
[0054] FIG. **27** illustrates how tables on the databases of various GVN devices are related to each other.
[0055] FIG. **28** illustrates the collaborative effort between various modules, mechanisms, technologies and other components of the GVN.
[0056] FIG. **29** illustrates the Advanced Smart Routing (ASR) feature of a GVN.
[0057] FIG. **30** illustrates building a series of encrypted tunnels between a Client (C) and a Server (S).
[0058] FIG. **31** illustrates the flow of information required by two peers in a peer pair.
[0059] FIGS. **32-35** illustrate the third Layer of the GVN with respect to neutrality and security of the GVN tunnel.
[0060] FIG. **36** Illustrates the weaving together of various network fabrics into a network tapestry.
[0061] FIG. **37** Communication pathways in a GVN for automated device collaboration FIG. **38** illustrates the problems and challenges of dynamic tunnel building.
[0062] FIG. **39** illustrates the bridging of two LAN s into a wide area network (WAN) via two or more EPDs.
[0063] FIG. **40** illustrates a Multi Perimeter Firewall Mechanism (MPFWM) running on a GVN.
[0064] FIG. **41** illustrates a GVN stack build over the top (OTT) of the Internet.
[0065] FIG. **42** compares the internet protocol IP stack, the OSI model, and the GVN network stack.
[0066] FIG. **43** illustrates global Internet flows between countries via many possible routes.
[0067] FIG. **44** compares the internet protocol IP stack, the OSI model, and the GVN network stack.
[0068] FIG. **45** illustrates a tunnel between two LAN s via the GVN.
[0069] FIG. **46** illustrates GNV layer one, layer 2, and layer 3 operations.
[0070] FIG. **47** illustrates the Advanced Smart Routing (ASR) feature and elements of the Geo-Destination Mechanism of a GVN within an End Point Device (EPD).
[0071] FIG. **48** illustrates examples of various concurrent types of paths for traffic to take via the GVN.
[0072] FIG. **49** describes the automated advanced smart routing (ASR) from one device to a second

device.

[0073] FIG. **50** illustrates the Secure Perimeter between the BB/Backbone layer Below Perimeter and the IP/Internet layer Above Perimeter.

[0074] FIG. **51** is a flowchart of Advanced Smart Routing (ASR) within a Global Virtual Network (GVN).

[0075] FIG. **52** is a flow chart of the various routes available through a GVN from an origin to a destination.

[0076] FIG. **53** is a flowchart of an algorithm governing the selection of traffic routing from a Start device to an End device.

[0077] FIG. **54** illustrates the modules required for automated device collaboration and information exchange in a GVN.

[0078] FIG. **55** illustrates the communications between EPD, SRV_CNTRL, and SRV_AP via the neutral API mechanism (NAPIM) of the GVN.

[0079] FIG. **56** illustrates various types of communications available between GVN devices via the NAPIM.

[0080] FIG. **57** describes API call groups between different types of devices within a Global Virtual Network (GVN).

[0081] FIG. **58** describes the steps taken for an API call from initiation on a client device through to sending to server device with a return back to client.

[0082] FIG. **59** is a flowchart outlining the interaction between the EPD and SRV_AP to achieve geographic destination functionality

[0083] FIG. **60** describes device collaboration within a geographic destination.

[0084] FIG. **61** illustrates how a globally distributed parallel file system (PFS) can operate within a GVN.

DETAILED DESCRIPTION

Overview

[0085] FIG. **1** shows a block diagram of technology used by and enabled by a global virtual network ("GVN") including the GVN core elements G**0**, GVN modules G**100**, and technology enabled G**200** by the global virtual network GVN. The GVN core includes an overview of the mechanism G**1** and its constituent component parts of Topology G**2** Construct G**3** Logic G**4** and Control G**5** layers. The GVN core G**0** also incorporates the relations to and with GVN Elements G**6**.

[0086] The GVN can include plug-in and/or stand-alone GVN modules G**100** including but not limited to: Neutral API Mechanism ("NAPIM") G**302** described in PCT/US16/12178; Geodestination ("Geo-D") G**104**, described in PCT/US15/64242, Advanced Smart Routing ("ASR") G**106**, Connect G**108**, and other modules G**110** described in U.S. Provisional Application 62/151,174.

[0087] The GVN also provides a platform which can enable other technologies including but not limited to: Network Tapestry G**202**, MPFWM G**204**; Network Slingshot G**206**, Network Beacon G**208**, Granularity of a tick G**210**, and other technologies G**212** These are described in in U.S. Provisional Application 62/174,394, U.S. Provisional Application 62/266,060.

[0088] GVN Modules (G**100**) and Technology (G**200**) enabled by GVN can operate on top of an existing GVN, as a component part of a GVN, or can be independent and utilize all or some isolated parts of a GVN to support their own stand-alone operations.

[0089] FIG. **2** show s a high-level block diagram of the Internet. The average user possesses a very cursory overview understanding of how the Internet functions. Host Source **2100** is the starting point and denotes a client device which could be a computer, a mobile phone, tablet, laptop computer or other such client. This client connects via the Internet **2200** to a host server **2300** to send or retrieve content, or to another host client **2303** to send or receive information.

[0090] A very non-technical user might assume that traffic to the host server follows path 2P **002** without even understanding that their data will transit through the Internet. Or they might think that traffic would flow via path **2**P **006** directly to another client device.

[0091] A user with some more understanding of how it works would understand that traffic flows via path **2**P **004** to the Internet **2200** and then via path **2**P **102** to a Host server Target **2300** or via path **2**P **104** to a Host (client) Target **2302**.

[0092] Users with some more technical knowledge will further understand that when sending an email, the this email will leave their client device **2100**, transit via path **2**P **004** to the Internet **2200** and then via path **2**P **202** to a mail server **2202**. Then the recipient of the email will make a request to retrieve the email via their host client **2302** along path **2**P **104** to the Internet and then down path **2**P **204** to the mail server **2202**.

[0093] This is about as detailed as the average person's understanding of the Internet gets.

[0094] FIG. **3** is a block diagram showing the resolution of Universal Resource Locator (URLs) to numeric Internet Protocol (IP) addresses via the Domain Name System (DNS).

[0095] A content request **3000** or push from host client (C) **3100** to host server (S) **3300** as files or streams or blocks of data flows from host client (C) **3100** to the host server (S) **330**M The response or content delivery **3002** is returned from host S to host C as files or streams or blocks of data. The host client device **3100** in a Client-Server (CS) relationship with host server (S) makes requests to access content from the remote host server (S) or sends data to remote host server (S) via a universal resource locator (URL) or other network reachable address.

[0096] The initial connection from the host client (C) **3100** to the Internet **3206** is shown as **3**P**02**—the connection from the host client (C) to a Point of Presence (POP) **3102** that can be directly facing. In other cases the host client (C) can be located in a local area network (LAN) which then connects to the internet via a point of presence (POP) and can be referred to as the last mile connection. The point of presence (POP) **3102** represents the connection provided from an end point by an internet service provider (ISP) to the internet via their network and its interconnects. This can be, but is not limited to, cable, fiber, DSL, Ethernet, satellite, dial-up, and other connections. If the URL is a domain name rather than a numeric address, then this URL is sent to domain name system (DNS) server **3104** where the domain name is translated to an IPv4 or IPv6 or other address for routing purposes.

[0097] Traffic from host client (C) **3100** to host server (S) **3300** is routed through the Internet **3206** representing transit between POPs (**3102** and **3302**) including peering, backhaul, or other transit of network boundaries.

[0098] The connection **3**P**04** between a POP **3102** and a domain name system **3104** used to look up a number address from a universal resource locator (URL) to get the IPv4 address or other numeric address of target server (S), can be directly accessed from the POP, or via the Internet **3206**. The connection **3**P**06** from a POP **3102** of an ISP to the Internet **3206** can be single-homed or multi-homed. Similarly, the connection **3**P**08** from the Internet **3206** to the remote ISP can also be single-homed or multi-homed. This connection is generally, to the ISP's or Internet Data Center's (IDC) internet-facing POP **332**. The connection **3**P**10** from the remote ISP's POP **3302** to the host server (S) can be direct or via multiple hops. The lookups from URL or hostname to numeric address via domain name systems is a standard on the Internet today and systems assume that the DNS server is integral and that the DNS server results are current and can be trusted.

[0099] FIG. **4** is a simplified illustration showing the upstream and downstream paths data takes from a host client device (C##) to another host client or host server device (S ##). The numbers used in the device label such as C**01** or S**08** are used for labeling purposes to locate individual devices and the number itself does not mean or imply that one device is larger or more powerful than another.

[0100] FIG. **4** shows Host Client Devices (C##), Host Server Devices (S ##), Switches (SW ##), Routers (R##), Regional Routers (RR##), Edge Routers (ER##), Core Routers (CR##). The communication path or pipe (P##) denotes a connection between two devices and the line thickness is to represent the size or bandwidth capacity of the pipe. The thinner the line, the lower the megabits per second (M bps). The thicker the line, the higher the amount of M bps or gigabits per second (Gbps). The distances of the P #are not to scale and the hop count or time to live (TTL) and latency or round-trip time (RTT) are not taken into account when noting the P ## between devices.

[0101] A simplified local area network (LAN) is downstream from the switch (SW) SW**01** It consists of wire line connection P**01** and P**04** to client devices C**01** and C**04** Wireless connections are described with dotted lines P**02** and P**03** between a wireless hub WLAN **01** and wireless client devices C**02** and C**03**.

[0102] The connection P**05** between the LAN and the point of present (POP) R**01** of their internet service provider (ISP) can also be referred to as The Last M ile. This POP R**01** is a hub which connects other spokes P**06**, P**07**, P**08** and P**09** to the corresponding switches of other clients such as SW**02**, SW**03**, SW**04**, and SW**05**. There is also an upstream path P**16** to a regional router (RR) RR**02**.

[0103] This hub and spoke topology is shown for POPs R**02**, R**03**, and R**04**, their spoke connections (e.g. P**10**, P**11**, P**12**, P**13**, P**14**, P**15**, P**51**, P**52**, P**53**, P**54**, P**55**, P**56**, P**57**, P**58**, P**86**) to their respective switches (e.g. SW**06**, SW**07**, SW**08**, SW**09**, SW**10**, SW**11**, SW**12**, SW**13**, SW**14**, SW**15**, SW**16**, SW**17**, SW**18**, SW**19**, SW**20**), and their connections (e.g. P**17**, P**18**, P**46**, P**28**) to their regional routers (e.g. RR**02**, RR**03**, RR**04**, RR**05**).

[0104] A further upstream connection P**19** from regional router RR**02** to edge router ER**02** describes the connection to the edge router of the ISP's network. Edge router ER**02** has a link P**20** to core router CR**03**. This can be thought of as the backbone of the internet. The link P**32** between CR**01** and CR**02** can describe a very large type of backbone called a backhaul network or when connecting various countries networks can be called international backhaul network.

[0105] The POPs R**01** and R**02** are both connected to regional router RR**02** and this can, but is not limited to, denote that these two POPs are located within the network of the same ISP.

[0106] For connectivity between devices within router R**01**'s networks and devices within router R**04**'s networks, the traffic will follow one of many possible paths such as P**16-**>P**19-**>P**20-**>P**3-**>P**31-**>P**24-**>P**27-**>P**28**. This likely describes the connectivity peering between the networks of two or more distinct ISPs and in the middle potentially other carrier peers depending on the owner of infrastructure that the traffic transits through. The traffic going through the backbone will be carried by the potentially highest capacity pipes. Traffic between router R**01** and router R**04** could also travel via path P**16-**>P**41-**>P**44-**>P**23-**>P**27-**>P**28**. While the path might seem shorter, this second path might be the least efficient because of the size of the pipes, the intermediary devices, peering relationships and the policies of a middle ISP, in control of edge router ER**03** for carrying the traffic between two other ISPs. There may also be choke points between them.

[0107] Another feature shown is in the connectivity of host servers S**08** to S**12** which are connected to switch SW**13**. This could be in an internet data center (IDC) or in a LAN. Switch SW**13** connects both to router R**03** via P**53** and to regional router RR**04** via P**46**. The connection P**46** may describe a leased line or direct digital connection for enhanced connectivity.

[0108] Another feature shown in this figure is that P**32** is as upstream as a path can get and individual host devices areas downstream as a path can go. Downstream from core routers CR**04**, CR**06** and CR**07** are edge routers ER ##connected to regional routers RR ##, which are connected down to the routers R ##located in POPs.

[0109] There may be other possibilities not described herein and in reality each router R ## has many more spokes to switches SW ## and there are extensively many more pipes P ## between devices. There may also be more layers of equivalent regional routers RR ## or edge routers ER ##devices or others in sequence.

[0110] FIG. **5** is a simplified illustration showing the boundary switches in the paths data takes from a host client device (C##) to another host client or host server device (S ##). This is very similar to FIG. **4** with one exception. On the backbone between core router CR**01** and core router

CR**02** at some point on the peering path between them, there are a series of boundary switches **400** which each have a limited capacity in relation to the backbone as a whole, and there can be congestion events through these switches.

[0111] FIG. **6** illustrates some example threats and problems which exist on the Internet. The network data paths have been simplified to describe an overview of connectivity with focus on threats from End Point Devices (EPD) and other threats from intermediary devices.

[0112] A request from a host client device C**002** to retrieve content from host server device **207** should follow the path P**109**->P**105**->P**103**->P**102**->P**101** and transit the Internet **101** to CP**01**->CP**02**->P**205**->P**207**. There may be a load balancer at the legitimate Internet Data Center (IDC) which may send traffic to a healthy host server **207** (via P**207**) or an infected host server **206** (via P**206**). The infected host server may send malware or viruses or other bad content back to the client device C**002**.

[0113] Another threat is the redirect of legitimate traffic to a spoofed host server **114**. Traffic should follow the path between C**02** and **207** as described above, however a spoofed server may syphon off legitimate traffic. The traffic will still follow a path such as P**109**->P**105**->P**103**->P**102**->P**101** and transit the internet **101** but instead of transiting to CP**01** on the way to the legitimate server, the traffic transits via P**113** to P**114** and is delivered to the Spoofed Server **114**.

[0114] A spoofed server may be designed to phish for confidential information or credentials or other data by looking like the real server to internet users. The average user may not be able to differentiate between a legitimate server and a spoofed server. Spoofed servers may also be utilized by a third party to block the delivery of legitimate traffic to clients by either sending null traffic back or changed content.

[0115] Public Domain Name Systems (DNS) servers are available on the internet to be queried by client devices to translate uniform resource locator (URL) such as a domain name www.thisdomain.com into a numeric IP address such as an IPv4 or IPv6 address so that the traffic from the host client device can find its way to the host server device.

[0116] If a DNS server such as **212** or **116** is poisoned **112** or spoofed **114**, the translated numeric IP address may be incorrect leading traffic to go to an illegitimate or compromised destination device. Another way DNS can be corrupted on the internet is if a device is improperly operating either not delivering results or by delivering incorrect results. Propagation of changes from master DNS registry servers to DNS servers also requires clear and valid connectivity or indexed results can become stale and inaccurate. An example of how to protect and secure DNS lookups is illustrated by a Secure DNS (DNSSEC) Server **110** and its connectivity via P**19**. This relies on the ability for client devices to connect to the DNS server **110** and for their handshakes to not be interrupted.

[0117] Even when both host client and host server devices are operating correctly, as the internet is not encrypted, there is a very real risk that a sniffer or interception device **204** interjected into a middle point within a communication path to a host server such as a mail server **203** could intercept and capture data. While traffic to the mail server **203** should flow from the internet **201** via P**202** to POP **202** to P**203** to the mail server **203**, a sniffer or interception device **204** will pull traffic via P**204** through it **204** and to P**222**. It is very hard to detect this kind of interference unless one specifically can identify the IP address of a hop in a communication path as belonging to a nefarious device rather than just another router as part of the infrastructure of the internet.

[0118] A growing threat is from BOT nets comprised of groups of infected devices **213**, **215**, **216** controlled by command and control (C&C) servers such as **234**. These devices can acting in unison to carry out bulk attacks such as distributed denial of service (DDoS) where host server devices can get overwhelmed by too many requests that flood their capacity resulting in the severing of requests from legitimate host client devices to either be slow or not resolve at all.

[0119] BOT nets can also be utilized to carry out stealthy hacking attacks under the coordination of C&C servers whereby the multitude of different source IP addresses trying dictionary password attacks are harder to completely block than the same attack would be if from a single IP address.

[0120] BOT nets are also a distribution mechanism for SPA M emails, phishing emails, malware distribution and other nefarious purposes.

[0121] National firewalls such as **304** are a hindrance to the free flow of information. These can either serve as censorship tools to block the flow of traffic which a country deems undesirable. It can also serve as an interception device to stealthily pilfer industrial, commercial or other secrets. Depending on the time of day, total internet traffic and health of these national firewalls, traffic transiting through them could experience a latency delay, or packet loss, or be shaped to a maximum bandwidth forming a bottleneck, or a combination of all of the above or even other problems.

[0122] The above noted example embodiments describe only some problems and threats. M any more exist and new threats crop up from time to time.

[0123] FIG. **7** illustrates Content Delivery Network (CDN) resolution and delivery of regionally specific content. Content Delivery Networks (CDN) can offer significant advantages in speed and flexibility and load balancing when serving content to clients. Content requests **7**REQ**000** flow from host client (C) **7100** to a host server (S) and the reply **7**RESP**002** flow of content delivery returns from the host server (S) to the host client (C) **7100** as files or streams or blocks of data.

[0124] The host client (C) **7100**, can be a device such as a laptop, desktop computer, phone, tablet, or other device that acts as a client in a Client-Server (CS) relationship with the host server (S). The host client (C) makes request(s) to access content served by a remote host server (S) via a universal resource locator (URL).

[0125] The POP **7102**, DNS server **7104**, Internet **7300** operate in the usual manner as described above.

[0126] In the case of CDN infrastructure, CDN Map Markers **7200** operates in coordination with CDN control server(s) **7202**. The CDN Map Markers **7200** and CDN control servers **7202** determine which region the host client device is located in and which CDN server the host client should connect to for content to be served. For example, If the host client **7100** is in Region A, it will be routed to the CDN server in Region A **7504** via the server's POP in Region A **7404**. Host clients **7100** in Region B will connect to a CDN server in Region B **7502** via the server's POP in Region B **7402**. Host clients **7100** in Region C will connect to a CDN server in Region C **7500** via the server's POP in Region C **7400**.

[0127] The initial CDN Map Marker **7200** lookup via **7**P**00**, via POP **7102**, via **7**P**004** may be very quick or could take a relatively high lookup time if the CDN Map Marker server is located in a region far from the client device. Once the lookup is done, traffic will flow to the nearest and or best available CDN Server via **7**P**008**.

[0128] For the sake of illustration of this figure, a region is defined as a geographic area which is different from another geographic area. It does not necessarily represent a large area but could be so and it also could represent a great distance from one region to another or they could be very close to each other. The key point is that clients in one region are to receive content via a CDN server from that region and not from another region.

[0129] In this example embodiment, the content for each region is different from the content of other regions. Between CDN servers **7500**, **7502**, and **7504** and the Origin Server **7600** are Content Regional Servers **7700**, **7702**, and **7704** which publish the regionally specific content to CDN servers in each region to be served to clients in their respective regions.

[0130] W hen a client **7100** in one region, for example region C, wants content served by a server **7502** or **7504** from another region, no matter what they do, they will only be served content from the server **7500** in their region. They cannot access other content even if they try to force it to connect to the content server in the region from which they desire to receive content. They keep being served content from their region without choice. Local DNS lookup **7104** resolves with IP pointing only to their region's CDN server **7500**. This may be due to a Global IP address which maps to only a CDN in their region (if global IP) or another reason. The result is that the client could be geo-blocked at **7**P**404** or **7**P**402**.

[0131] Normal connection via **7**P**008** based on current Geo-Location is not subject to blocking and traffic flows so that host client **7100** receives content for that Geo-Location via host server **7500**.

[0132] For targets different from Current Geo-Location **7502** and **7504**, traffic is stopped at **7**P**402** and/or **7**P**408** and the host client is denied content from the remote geo-destination(s). They may be forced to connect to the server in their current location **7500** or receive nothing or an error message or just undesired content depending on the configuration and policy of the CDN control system **7202**.

[0133] FIG. **8** illustrates the operation of a proxy server. Content requests or pushes **8**REQ**000** flow from host client (C) to host server (S) as files or streams or blocks of data. Content delivery **8**RESP**002** is returned from host server (S) to host client (C) as files or streams or blocks of data. Host client **8100** a client device in Client-Server(CS) relationship with host server **8500** makes request to access content from the remote server (S) via a universal resource locator (URL). This request goes through a gateway (GW) device **8102** running proxy client software. In other cases, the proxy client software can be running directly on the host client **8100**. The proxy client software connects to a Proxy Server **8306** via a tunnel, encrypted or unencrypted, via path **8**P**02** from the gateway GW **8102** to point of presence (POP) **8200**, via path **8**P**04** to WAN **8308** (part of the Internet), via path **8**P**6** to the Proxy Server **8306** in a remote region. The traffic egresses from the proxy server **8306** via path **8**P**16** into the open internet **8300** and connects to host server **8500** in target region via path **8**P**12** to POP **8302**, and then via path **8**P**10**.

[0134] The host server views the traffic as coming from the IP address and geo-location of the proxy server. If this IP is in the same region as defined by the server in the target region, the desired content will be served. To aid in this localization, proxy servers will usually connect to DNS servers **8404** in the same region as the proxy server is located.

[0135] FIG. **9** illustrates a point-to-point tunnel TUN built between two gateway devices **9**A**1** and **9**B**1** Each device **9**A**1** and **9**B**1** is at the edge **9**EDGE-**1** and **9**EDGE-**2** between the internet EH**3** through EH**15** and their corresponding local area network (LAN) **9**A**2** and **9**B**2**.

[0136] The baseline from EH**1** through EH**17** describes the number of hops from point to point. The number of hops from EH**3** to EH**15** is hypothetical and provided for illustrative purposes and may be more or less in real world connection paths. The number of hops that clients utilizing the tunnel **9** TUN from **9**A**2** to **9**A**1** to **9** TUN to **9**B**1** to **9**B**2** will be approximately four or five visible hops.

[0137] This example embodiment describes a scenario where the LAN **9**A**2** connects through their gateway **9**A**1** to the network of one internet service provider **9**ISP-**1** and where LAN **9**B**2** connects through their gateway **9**B**1** to another internet service provider **9**ISP-**3**. This example embodiment further illustrates that **9**ISP-**1** does not peer directly with **9**ISP-**3**. Both **9**ISP-**1** and **9**ISP-**3** have a requirement that their network traffic in both directions must transit through the network of another internet service provider **9**ISP-**2**. Interconnection between **9**ISP-**1** and **9**ISP-**2** is defined as peering point **9**PP-**01** and from **9**ISP-**3** to **9**ISP-**2** as **9**PP-**02**.

[0138] The point of this example embodiment is to illustrate that over the internet, it is common for a third-party internet service provider or equivalent such as a backbone or backhaul provider to carry the traffic of other internet service providers. There is little to no control by **9**ISP-**1** or **9**ISP-**3** over how **9**ISP-**2** will carry their traffic. While customers **9**A**2** of **9**ISP-**1** are able to directly complain about service issues to their provider **9**ISP-**1** and **9**B**2** can complain directly to **9**ISP-**3**, if the issue is with **9**ISP-**2** then there is very little that **9**A**2** or **9**B**2** can do to directly influence **9**ISP-**2**.

[0139] Potential congestion points can occur on any device but **9**PP-**01** and **9**PP-**02** are areas of concern as they are peering points. There is limited control over routing and quality of service of the total connection. As a consequence it may be difficult for point-to-point tunnels to maintain a high-quality, stable connection over distance, especially when there are portions of traffic transiting third party networks.

[0140] FIG. **10** shows the relationship of security features between device scope **1080** and system-wide scope **1090**. It also notes communications scope **1098** and device collaboration **1089**.

[0141] With respect to device scope **1080**, the GVN protects client privacy of their data, network data flow, credentials, peer pair information, as well as protecting the physical device from hacking, the proprietary code contained therein from tampering or theft, and other threats.

[0142] System-wide scope **1090** entails protection from hacking or other offending traffic such as DDoS attacks, guards against malfunction, does routing around sub-optimal devices or paths, balances and spreads load, and protects against the running out of resources, IP addresses, or other global issues.

[0143] Communications scope **1098** focuses on the pathways for traffic push through the GVN predominantly through traffic tunnels TUN. It also covers the egress ingress points (EIP) between external networks and the internal network of a GVN. Protects from stream hijacking, man-in-the-middle attacks, poisoned information sources (such as bad DNS, etc.), and other threats. Furthermore, testing of quality of various network segments and their properties allows for the GVN to be able to understand complete path QoS and to route around problems.

[0144] The device collaboration **1089** security features are in place to protect the operational integrity of the various devices within a GVN. Secure back channel, anti-hacking mechanism, DNS safety net, various protections of Db such as rotating keys, neutral API mechanism (NAPIM), automated testing, updating, peer pairs relationships, validation and other modules ensure that system integrity is maintained.

[0145] FIG. **11** illustrates the information flow between devices of a Global Virtual Network. A central repository comprised of database B**200** and file storage HFS**200** resides on a Central Server (SRV_CNTRL) **200**.

[0146] Communication paths between devices labeled P ###can represent an API call, database replication, direct file transfer, combination such as database replication through API call, or other form of info exchange. The thicker lines of **11**P**200100**, **11**P**200300**, **11**P**200500**, **11**P**100200**, **11**P**100300**, **11**P**10011500**, **11**P**300200**, **11**P**300500**, and **11**P**500200** represent communications between GVN devices which have a peer-pair and therefore privileged relationship with each other.

[0147] There is circular pattern of peer-pair communication illustrated from SRV_CNTRL **200** to EPD **100** via **11**P**200100**, to SRV_AP **300** via **11**P**200300**, or to other devices **11500** via **11**P**200500**. The EPD **100** communicates with SRV_CNTRL **200** via **11**P**100200**, SRV_AP **300** via **11**P**100300**, and other devices **11500** via **11**P**1001500**.

[0148] In some instances, there will be a loop of information shared between devices such as in the case when an EPD **100** may request information via **11**P**100200** from SRV_CNTRL **200** which is sent back to EPD **100** via **11**P**20010**.

[0149] In other instances, one device may report information relevant to other devices such as an SRV_AP **300** reporting via **11**P**300200** to SRV_CNTRL **200** which is then sends information via **11**P**200100** to EPDs **100** and SRV_APs **300** other than the reporting SRV_AP **300** via **11**P**200300** and to other devices **11500** via **11**P**200500**.

[0150] In yet other instances a full loop is not required such as the sending of log information from a device such as an EPD **100** to SRV_CNTRL **200** via **11**P**100200**, there is no need to further forward this information onward. However, logging information may at a later time be moved from repository on SRV_CNTRL **200** to a long-term log storage server **11500** via **11**P**200500**.

[0151] Direct link **11**P**100300** is between devices EPD **100** and SRV_AP **300**. Direct link **11**P**300500** is from SRV_AP **300** to other devices **1150**. Direct links involve communications between devices which do not need involvement of SRV_CNTRL **200**.

[0152] The PUSH info from SRV_CNTRL **200** could bean RSS feed or other type of information publishing via **11**P**306**. The API from SRV_CNTRL **200** could be either a traditional API transaction or RESTful API call with request made via **11**P**302**REQ and response received via

11P**302**RESP. The PUSH info and API elements are presented to illustrate devices which do not share peer-pair relationships, privileged status, and or similar systems architecture with GVN devices.

[0153] FIG. **12** describes the stack for supporting the automation of some devices in a GVN. In particular this figure shows the modules required for automated device collaboration and networking plus O/S management

[0154] EPD **100** is the endpoint device. SRV_AP **300** is an access point server which is located in the target destination region. SRV_CNTRL **200** is a central control server accessible by both the EPD and the SRV_AP as well as by other devices which may support a graphic destination mechanism, or other GVN module, component or server.

[0155] Each device EPD **100**, SRV_AP **300** and SRV_CNTRL **200** stores information about itself in a local information repository in the form of lists, files, database tables and records, and other means. This repository also contains information about peer device relationships, stores logs, plus other relevant operational information. The SRV_CNTRL **200** also has additional storage functionality and its role is to provide information to other devices relevant to them and or to the peer devices which they may connect with, to evaluate current conditions and provide centralized control-like guidance such as the publishing of a server availability list and other functionality. A neutral API mechanism (NAPIM) can send info between devices and the peers which they connect with, and can also be used to update the API itself.

[0156] The database S**293** on the SRV_CNTRL **200** acts as a repository for information about itself as well as a centralized repository for other devices. There can be many different SRV_CNTRL **200** servers acting as multiple-masters in many locations. Each database can store certain information including tunnel information, peer information, traffic information, cache information, and other information. Security and other aspects are independently managed by each device including heartbeat functionality, triggered scripts and other mechanisms.

[0157] GVN software D**196**, D**296**, D**396** includes tunnel builder/manager, virtual interface manager, automated smart routing, test modules, security, logging, and other functionality. FIG. **11** also shows operating system (O/S) level packages D**195**, D**295**, D**395** and includes, hardware and software drivers, drivers, installed packages including their dependent software packages, and other items built on top of the hardware components of the system.

[0158] FIG. **13** illustrates GVN topology, including a backbone segment over internet or dark fiber. International Patent Application No. PCT/US15/64242 SYSTEM AND METHOD FOR CONTENT RETRIEVAL FROM REMOTE NETWORK REGIONS, discloses a feature where multiple files are clumped together into a larger file to be sent by a file transfer via “chained cache” from one geographic region to another geographic region. For this feature to be advantageous, the file transfer needs to be as fast as possible. As a transport for a clump of various data payload “files”, the information slingshot method of this invention moves a larger block of data faster from one end of the world to the other than methods of prior art.

[0159] Referring to FIG. **13**, there are multiple zone shown: LAN zone 0 (ZL**00**), LAN zone 1 (ZL**10**), Internet zone 0 (ZI**00**), Internet zone 1 (ZI**10**), Internet zone 2 (ZI**20**), Internet zone 3 (ZI**30**), Internet data center zone 2 (ZD**20**), and Internet data center zone 3 (ZD**30**).

[0160] SRV_BBX **1372** in region or zone ZD**20** can be connected to SRV_BBX **1380** in another region or zone ZD**30** via a dark fiber connection **13P220** over dark fiber **13220**. SRV_BBX **1372** directly writes a file to parallel file storage PFS **1382** via remote direct memory access (RDMA) over **13**P**220** bypassing the stack of SRV_BBX **1380** via path **13**P**82**. SRV_BBX **1380** uses this invention to directly write a file to parallel file storage PFS **1374** via remote direct memory access (RDMA) over **13**P**220** bypassing the stack of SRV_BBX **1372** via path **13**P**74**.

[0161] Path **13**P**210** can be IPv4 or some kind of standardized internet protocol over which traffic flows from SRV_AP **13300** to and or from SRV_AP **13310** via path **13**P**210** over-the-top of the GVN via a tunnel or other type of communication path.

[0162] This illustrates that various types of network fabrics can be combined into a greater network tapestry. These fabrics can be seamlessly woven together as described in U.S. Provisional Patent Application No. 62/174,394. This can be either a stand-alone method or can be integrated as a network segment within a greater network path comprised of various network segments. This example embodiment illustrates the topology of a global virtual network (GVN), its various devices, communications paths, and other embodiments. It shows how various geographic regions or zones or territory are linked together over various types of paths.

[0163] FIG. **4** illustrates a distributed firewall (FW) in the cloud enabled by a GVN. Because of the nature of a GVN's topology, device-to-device communications, and secure traffic path, a firewall mechanism can be cloud based and also can be virtualized. With a firewall facing hop **144** within the flow to and from the GVN via an egress ingress point (EIP) to the open internet **14000**, there can be a cloud firewall (CFW) load balancer **144**LB which will be able to allocate cloud firewall resources such as **144-2**, **144-3**, and so on.

[0164] This on demand scalability offers many advantages to clients of a GVN. By absorbing the hits of the attacks for incoming threats in the cloud, the client's last mile connectivity is not affected. This cloud firewall combined with a control node and analyzer allows for the FW in the region under attack to be aware of the nature, source, signature and other features of the attack so that it can be aware and be prepared to thwart the attack if the target shifts. Furthermore, information about past and current attacks can be shared via the neutral API mechanism (NAPIM) of the GVN to other CFW instances, so that global threat awareness is possible. This also offers advantages of running simultaneous types of FW mechanisms as discussed with respect to FIG. **15**.

[0165] FIG. **15** illustrates a multi-perimeter firewall (MPFW) in the cloud empowered by a global virtual network. The GVN tunnel **15**TUN**0** is over the top (OTT) of the internet between an end point device (EPD) **15100** and an access point server (SRV_AP) **15300** in close proximity to the EPD **15100**.

[0166] The three perimeters indicated in this example embodiment are **15**M**1** which denotes the boundary between a client location and their link to the internet, **15**M**2** which is a boundary in the cloud at a datacenter in close proximity to SRV_AP **15300**, and **15**M**3** which is another boundary at either the same data center as SRV_AP **15300** or at another location in close proximity to SRV_AP **15302**.

[0167] The tunnel **15**TUN**2** is similar to **15**TUN**0** and different in one respect in that it connects a personal end point device (PEPD) **15130** which can be mobile and therefore connects to SRV_AP **15300** through public access wireless or wired or other networks to integrate into the GVN.

[0168] Each SRV_AP **15300** and SRV_AP **15302** may represent one or more SRV_AP devices through which the EPD **15100** and/or EPD **15130** may concurrently connect with via one or more multiple tunnels.

[0169] There are three types of firewall described in this example embodiment. FW local **15442** is an example of a firewall which a client may use to protect their local area network (LAN) from internet based threats. This is typically located between the EPD **15100** and the LAN **15000**. This FW **15442** may offer features such as IP address and port blocking, forwarding, and other functionality. The other two types of firewall illustrated are FW SPI **15446** located at **15**M**3** which is offers Stateful Packet Inspection (SPI) and FW DPI **15444** located at **15**M**2** which provides Deep Packet Inspection (DPI).

[0170] The difference between SPI and DPI has to do with a tradeoff in performance versus visibility. SPI examines at the headers of packets to look for malformed information, or for patterns, or to match IP address or port or other information from its list of known threats against the current flow of packets. DPI as its name implies takes a deeper look at the whole packet and in the case of a multi-part, multi-packet transmission, will look at the compilation of a series of a packets to gain insight into the data being transferred.

[0171] All firewalls can be configured to investigate and apply rules to both incoming and outgoing traffic, and provide other related functionality. In many cases, clients will have to choose between the efficiency of SPI vs. the thoroughness but resource and time intensive requirements of DPI.

[0172] A GVN offers the opportunity to distribute these FWs at various points in the cloud. And for the various types of firewall to be operating in lockstep with each other, without impeding the flow of traffic.

[0173] By locating FW SPI **15446** at **15**M**3**, the closest edge to the Internet **15302** via EIP remote **15310** the bulk amount of attack traffic from known source IP addresses or with recognized malicious headers can be thwarted. Traffic flow from SRV_AP **15302** to FW SPI **15446** via **15**T**10** and back via **15**T**12** FW SPI **15446** can be a CFW load balancer (see FIG. **14**) which has plenty of resources on demand. SRV_AP's at **15**M**3** can be on a multi-honed backbone with huge capacity. Therefore, at this first perimeter, attacks can be caught, protecting bandwidth within the GVN.

[0174] At the next perimeter **15**M**2**, the FW DPI **15444** can have all traffic flow through or just receive a copy of traffic via **15**T**20** from SRV_AP **15300** and it may or may not return traffic via **15**T**22** The key point is that the DPI feature can be a trailing indicator allowing certain traffic through but analyzing and recording the results. This FW DPI **15444** can also be a CF W which is load balanced with resources available on demand as needed to cope with large scale events when needed without individual clients having to administer or bear the cost burden for maintaining the infrastructure during normal times.

[0175] The information from FW SPI **15446** and FW DPI **15444** is shared with each other via internal communications path **15**P**6** which may be carried by the NA PIM of the GVN, or through GVN tunnel, or GVN back channel, or via other communications pathway(s). Each FW mechanism also shares information with the central, control servers (SRV_CNTRL) **15200** of the GVN. This information can be relayed to other FW SPI and FW DPI around the world so that attack vectors, sources, payloads, and other related information can be made available in a database so that SPI and DPI checks can have a point of reference to check against. This permits more efficiencies of scale as the global distribution of information provides an added safety net.

[0176] The catching of offending traffic outside of a client LAN and in the cloud protects the client's last mile internet connectivity from being saturated by unwanted traffic. Offloading of traffic to CFW which are scalable also offers many advantages to clients.

[0177] The FW local **15442** may be a standalone device, a software application (APP) running inside of the EPD **15100**, or other kind of FW device.

[0178] The FW SPI **15446** and FW DPI **15444** devices and related devices such as load balancers, cloud firewalls, or other devices may be custom made or can be off the shelf provided by other vendors offering best of breed options to clients. These devices must be able to receive and forward traffic, identify threats and most importantly to be able to communicate their threat findings and to receive threat profiles and other information from other devices.

[0179] As the threat data accumulates, analysis can be made of the content, the patterns, the attack vectors, and other information gathered by the FWs. This analysis can provide a basis through which heuristic analysis can be applied to new potential threats.

[0180] This can only be achieved by the secure network optimization (SNO) services of a GVN or similar network which consists of related devices connected both by secure tunnels and communication paths.

[0181] FIG. **16** illustrates a logical view of the software architecture of three types of network devices working together as a part of a Global Virtual Network (GVN). As shown, the software and hardware can be distributed within the network devices and across different circuit boards, processors, network interface cards, storage, and memory.

[0182] One described network device is an End Point Device (EPD) **100**. Another described network device is Central Server (SRV_CNTRL) **200** and the third device is an Access Point Server (SRV_AP) device **300**.

[0183] The EPD **100** is connected to the SRV_AP **300** via encrypted tunnel described by communication path could be encrypted tunnel SYSC**04** to a Point Of Presence (POP) SYS**406** through communication path SYS**06** to a WAN SYS**400** to communication path SYSCP**10** to POP SYS**402** to communication path SYSCP**12** The path transiting WAN SYS**400** could also be through the regular non-encrypted internet.

[0184] Each device EPD **100** and SRV_AP **300** can also connect to the SRV_CNTRL device **200** via communication path SYSCP**08**.

[0185] The software architecture of EPD **100** and SRV_AP **300** are very similar to each other with the differentiation by role of each device in their operations, and some differing modules.

[0186] The lowest level of each device are the memory (RAM) **106**, **206**, **306** and processors (CPU) **102**, **202**, **302**, and the network interfaces (NIC) **108**, **208**, **308**. All of these are on the hardware level. The operating system (O/S) **110**, **210**, **310** can be a LINUX system or equivalent system such as Debian or other. This description of an operating system includes packages and configuration for routing, hosting, communications and other system level operations software.

[0187] On top of the operating system **110**, **210**, **310** exists a system software layer **112**, **212**, **312** of the Global Virtual Network's (GVN's) operating systems. Operating here are custom commands, system modules, managers and other constituent parts, as well as other components of the GVN. Each type of device of the GVN may have some or all of these portions of the system software layer or different portions depending on their role.

[0188] Database modules Db **120**, **220**, **320** and Hosting Modules **122**, **222** and **322** are configured in this example embodiment for the listening, sending, processing, storage, retrieval and other related foundation level operations of the GVN's neutral API mechanism (NAPIM), graphic user interfaces (GUI) and other server side script hosted sites. Database **120**, **220**, **320** (Db) modules could be M ySQL or equivalent such as MariaDb and hosting modules **122**, **222** and **322** could be Apache and PH P scripting or other type of hosting languages. Command Line scripts are also used and can be written in Bash, C, PHP, Pearl, Python or other language.

[0189] Billing modules can collaborate and share information such as the amount of data consumed by tunnel traffic to be billed by a consumption model. The accounting module ACC **132 232 332** operates on the EPD **100** and the SRV_AP **300** has a corresponding Billing module. Both can provide financial information to report screens, payment forms, emailed statements and other financial data produced by the GVN.

[0190] SRV_CNTRL **200** has a Repository Manager **238** which handles billing info, tunnel manager information and other data which can be utilized by various devices within the GVN. The Repository Manager **238** also handles the coordination of the sharing of peer pair info, credentials and other information to individual devices connecting to other API peers via the neutral API mechanism (NAPIM) of the GVN.

[0191] The EPD **100** has an API Module **130**, SRV_CNTRL has API Module **230** and the SRV_AP **300** has an API Module **330**. For the simplicity of explaining this example embodiment only one API module has been expressed per device. In fact, devices may have a combined client and server role depending on its function within the GVN.

[0192] A Cache Manager on SRV_CNTRL **200** manages the master index of various chained caches distributed across many devices of the GVN. The Compression Engine **136** on the EPD **100** and **336** on SRV_AP **300** manages the compression and decompression of data both stored on files, in DB tables or for streaming transport data.

[0193] Advanced Smart Routing (ASR) **150** module on EPD **100** handles the routing of traffic from an EPD **100** to the best egress point for destination via routes of the GVN.

[0194] Remote Fetcher BOT **331** on the SRV_AP **300** is a core component of the Geo-Destination Mechanism (Geo-D).

[0195] DNS Manager **254** on SRV_CNTRL **200** manages the master DNS index which can seed DNS Servers on various GVN devices, such as DNS **154** on EPD **100**.

[0196] A Logger Manager on SRV_CNTRL **200** manages both local logs and logs shared by devices to the Repository via API calls. The Logging Manager in this example embodiment imbues the functionality of recording operational events, API actions and transactions and the Logger also has other roles and processes for various aspects of the GVN operations.

[0197] Local Cache **152** on EPD **100** and local Cache **352** on SRV_AP **300** cache data locally.

[0198] GVN Managers **272** operate on SRV_CNTRL **200** to control the operations of various components of the system both on the SRV_CNTRL **200** and other devices of the GVN.

[0199] Local DNS server and cache **154** on EPD **100** and **354** on SRV_AP **300** allow for caching of DNS lookups for fast, local retrieval. The DNS **154** and **354** can be completely flushed, individual items purged, or timeouts set for retrieved lookups to be deleted after a certain period of time has transpired.

[0200] On the EPD **100** is a Content Delivery A gent (CDA) **158** which is a component of Geo-D.

[0201] On the SRV_AP **300** is a Content Pulling Agent (CPA) **358** also a component of Geo-D. The CPA **358** works with the BOT **311** on SRV_**300** to pull content from a distant region using local DNS **354** seeding from that Region. The CPA **358** sends fetched content to the CDA **158** utilizing tunnels, caches and other improvements of the GVN.

[0202] A firewall (FW) (not shown) on EPD **100**, on SRV_CNTRL **200** and on SRV_AP **300** operates to protect access to both the device and the communications paths between the device and others.

[0203] Connectivity Manager (not shown) on EPD **100** and on SRV_AP **300** manages the tunnels between devices and other device to device communications paths. Compression Manager on **215** of SRV_CNTRL **200** manages compression both locally and also coordinates with Compression Engines **136** on EPD **100**, **336** on SRV_AP **300** and on other devices of the GVN. Routing on EPD coordinates with ASR **150**, Geo-D, and other elements to manage traffic routing.

[0204] The structure of the database tables in SDB **100**, SDB **200**, and SDB **300** are equivalent for device operations while the data for each is specific for device types, and each device has identity specific devices. On SRV_CNTRL **200**, the Repository Database SDB **202** is where unique information is stored for all devices and this information can be used by the Repository Manager **238** to communicate API credentials, tunnel info, or other information to a device.

[0205] Stored within each device database are identity and API peer info about the device itself and its peer pair partners, transaction lists and queue data, and other information. There are other uses for the described methods and databases beyond what is described but for simplicity of illustration, this example only covers a few example core functionality elements.

TOPOLOGY

[0206] FIG. **17** illustrates a GVN using hub and spoke topology with a backbone and octagon routing. FIG. **17** shows the network topology of a GVN in two different regions **17**-RGN-A and **17**-RGN-B and how the regions are connected via paths **17**-P**0**A and **17**-POB through global connectivity **17**-RGN-ALL. In addition, FIG. **17** shows the hub & spoke connections in each of the two regions. FIG. **17** is similar to FIG. **15** and adds multiple egress-ingress points (EIP) in each region as added spokes to the hub and spoke model.

[0207] SRV_BBX **17-280** and SRV_BBX **17-282** are backbone exchange servers and provide the global connectivity. SRV_BBX may be one or more load-balanced servers in a region serving as global links. Access point servers (SRV_AP) **17-302**, **17-304** and **17-306** in **17-17**-RGN-A connect to SRV_BBX **17-280**. The central, control server (SRV_CNTRL) **17-200** serves all of the devices within that region and it may be one or more multiple master SRV_CNTRL servers. End point devices (EPD) **17-100** through **17-110** will connect with one or more multiple SRV_AP servers through one or more multiple concurrent tunnels.

[0208] This figure also shows multiple egress ingress points (EIP) **17**-EIP**420**, **17**-EIP**400**, **17**-EIP**430**, and **17**-EIP**410** in each region as added spokes to the hub and spoke model with paths to and from the open internet. This topology can offer EPD connections to an EIP in remote regions routed through the GVN. In the alternative this topology also supports EPD connections to an EIP in the same region, to and an EPD in the same region, or to an EPD in a remote region. These connections are securely optimized through the GVN.

[0209] FIG. **18** illustrates the backbone connections between some GVN Global Nodes and their corresponding service areas in North America, Europe and Asia. As described in the Legend box at the bottom right of FIG. **18** each zone noted herein from a networking perspective is described as a Global N ode. Global Nodes are connected to each other via High Performance Network links. The lower the latency between the points, the faster information can be transferred.

[0210] A round the Global Node two rings denote the type of connectivity quality zone in for example a radius from the center where the source info is located. This is for simplification only as many factors determine the size and shape of these zones. However, the two zones can be differentiated from each other as the closest one being a High Performance Zone and the other being an Optimal Service Area.

[0211] The farther a querying client or server or other type of device is from the global node, the longer it takes for information to flow and at some point the distance is so great that the QoS reduction is such that the device is no longer in the High Performance Zone, but is now located in the Optimal Service Zone.

[0212] If the QoS drops below a certain threshold, the device is located outside of the Optimal Service Area and therefore the distance between it and the global node is so great that the advantages offered by a GVN, with the exception of security, may be a moot point.

[0213] FIG. **18** shows zones SJC **18-01** for San Jose, CA, USA, JFK **18-02** for New York, NY, USA, AMS **18-11** for Amsterdam, NL, NRT **18-21** for Tokyo, Japan, and HKG **18-22** for Hong Kong, SAR, China. There are many other reasonable locations around the world within which to place a global node that are significant, but for simplicity of illustration only a few are shown for illustrative purposes.

[0214] FIG. **18** also show representative paths between each global node, for example such as between JFK **18-02** and AMS **18-11**. In reality, there are a multitude of paths representing undersea cables between the two points.

[0215] FIG. **19** illustrates the connectivity between various devices within a GVN noting multiple connection paths to hub devices from devices in the spoke(s). The placement of SRV_BBX (backbone exchange server) **19-800** and **19-810** points are based on the client's location relative to the to best Internet Data Center (IDC) with respect to pipes, interconnects to serve a target region while connecting global locations via paths **19**-BB**2** and **19**-BB**6**.

[0216] The SRV_BBX acts as a hub for the region it serves. Hubs are connected to each other by Tunnels over the top (OTT) of Ethernet links in the Internet, Tunnels over direct Ethernet links, Infiniband over Fiber, Infiniband over Ethernet, or other form of connectivity between regions. Each hub serves various SRV_AP servers such as **19-302**, **19-306**, **19-308** serving one area within a global region. SRV_AP **19-312**, **19-316**, and **19-318** can serve another area of a global region.

[0217] End point devices (EPD) such as **19-100** through **19-128** will connect with the most appropriate SRV_AP server relative to their location, their network connectivity, its peering and other relevant factors. These factors are constantly changing and therefore multiple tunnels to multiple SRV_AP servers are maintained by an EPD at all times. Each EPD connects with various (one or more) SRV_AP servers simultaneously.

[0218] There are egress ingress points (EIP) at EPDs, at SRV_APs and at other locations where traffic can leave or enter the GVN to or from the internet with the GVN securing and optimizing the traffic as far as possible.

[0219] SRV_AP devices such as SRV_AP **19-9308** and SRV_AP **19-318** may also have connectivity to each other through a tunnel path such as **19**P**60** so that two EPDs such as EPD **19-110** could connect with EPD **19-328** via path **19**P**22** to **19**P**60** to **19**P**58**.

[0220] The central, control server (SRV_CNTRL) **19-200** is linked to various devices via paths such as **19**P**62** to SRV_AP **19-302** for neutral API mechanism (NA PIM) information exchanges.

The EPDs also connect with the SRV_CNTRL **19-200** via NA PIM paths. To keep this example embodiment relatively simple, the NA PIM EPD to SRV_CNTRL paths were not shown.

[0221] The NAPIM information exchanged between the SRV_CNTRL and various devices can be used to share usage statistics, tunnel building information such as IP addresses, ports, protocols, security credentials, certificates, keys, and the sharing of other information to allow for the automated and secure operations of the GVN.

[0222] FIG. **20** illustrates how GVN modules and devices interact. A global virtual network (GVN) consists of various devices which operate independently as well as in collaboration with other devices. While the role of each may be different based on their type and its primary function, they follow similar code base, database schema and other architecture elements.

[0223] Infrastructure is installed in a region to support the operations of EPDs and PEPDs. Devices such as End Point Devices (EPD) **100**, Portable End Point Devices (PEPD), and End Point Hubs (EPH) connect various LAN, PAN and other networks to the GVN via tunnels to Access Point Servers (SRV_AP) **300**. Each device has its own locally hosted databases.

[0224] Redundancy is provided by multiple servers of each type per region with multiple master SRV_CNTRLs and other server types. The central data repository is located on the Central, Control Server (SRV_CNTRL) **200**. The SRV_CNTRL's job is to connect to various devices via the neutral API mechanism of the GVN. The API calls via the NA PIM of the GVN are via paths **20**P**02** for communications between devices such as EPD **100** to SRV_BC **20-502**. Device_ID and registration/regional mapping in the Db repository on SRV_CNTRL allows for API peer pair relationship management, generates appropriate Server Availability List (SAL), and accepts logging. This allows for the efficient management of the relationships and connections with SRV_AP and GW servers.

[0225] The backend servers and infrastructure devices of the GVN include Back Channel Servers (SRV_BC) **20-502**, Secure Boot Servers (SRV_SB) **20-504**, Authentication, Authorization, Accounting Servers (SRV_AAA) **20-508** and Logging Servers (SRV_LOG) **20-516**, and others.

[0226] The gateway servers and other devices are connected to the SRV_CNTRL **200** via connector **20**AD**0** and to gateway devices via All Devs hub **20**A D**2** This can include a gateway email server (SRV_GW_Mail) **20-510**, a gateway server for financial transactions (SRV_GW_FIN) **20-518**, and/or a gateway server for Third Party Connections (SRV_GW_TPC) as one type of other the SRV_GW_***20-512**.

[0227] Gateway servers operating with a specific role can be tweaked for that functional role and secured in such a way to protect them. By delegating an email gateway server, it can be set up as a secure email sender and receiver. This will require configuration and maintenance and observation of its operations. But at the same time, no other servers would need to handle email freeing up an administrative burden for those devices. All devices can forward emails via data payload sent to API by action call to request email to be sent. Flags in the payload can indicate whether email should be sent right away or at a specific time or at which priority. Other settings can govern how it is sent. SRV_GW_EMAIL will receive these data payloads, add them to its Email sending queue and email manager will handling when and how to deliver the email and will accordingly log the event. Bounce backs, replies, and other incoming emails can also be handled by the one point server type SRV_GW_EMAIL.

[0228] Logging servers and other devices can also be reached by GVN devices via **20**AD**4**.

[0229] FIG. **21** illustrates additional details of how GVN modules and devices interact. The additional details include communications paths such as **21**Q**00** from SRV_BC **4-502** to SRV_CNTRL **200** for reporting of information from back channel server to central, control server. The key point is that while a GVN device will need information about itself, its peers, its connectivity options and other information in order to operate, the sharing of performance and other data to SRV_CNTRL **200** and or other devices allow for an overall perspective of the greater system. The constant feedback loops allow for automated adjustments and learning-on-the-fly for better decisions to be made.

[0230] FIG. **22** illustrates the topology and connectivity of GVN modules and devices and how they interact with other devices on the Internet. Communication paths shown in FIG. **22** include Path External (PE), Path of Tunnel (for traffic) (PT), Control Path (CP), Encrypted System Path (ES) and Path for API communication between devices of a GVN (PA), and more.

[0231] The Central Server (SRV_CNTRL) **200** contains a repository of files and databases holding important system information. The SRV_CNTRL is able to connect with all GVN devices via PA paths for API communication. An End Point Device (EPD) **100** is the network access point between a local area network (LAN) and the Internet via various concurrent potential communication paths.

[0232] Advanced Smart Routing (ASR) within an EPD can send local traffic to the Internet **22-010** closest to it via **22-**PE**00** to a Point of Presence (POP) **22-020** to **22-**PE**02** The Back Channel Server (SRV_BC) **22-502** connects to the EPD **100** via a back channel connection from **22**ES**04** through **22-010** via **22**ES**02** to **201** to **22**ES**020** into the EPD **100**. The ES ##paths are encrypted Control Paths and are independent of traffic carrying Paths of Tunnels.

[0233] The EPD **100** maintains multiple tunnels to each of multiple Access Point Servers (SRV_AP) via **22**PT**00** and **22**PT**02** to SRV_AP **300**, via **22**PT**04** and **22**PT**08** to SRV_AP **22-302**, via **22**PT**10** and **22**PT**12** to SRV_AP **22-306**, and via **22**PT**14** and **22**PT**16** to SRV_AP **22-308**.

[0234] This figure is not to scale but for example, SRV_AP **22-302** and SRV_AP **300** are in the same region and egress from the GVN to the Internet **22-032** via **22**PE**04** to POP **22-022** to **22**PE**08** to the Internet **22-012** and via **22**PE**16** via POP **22-026** to **22**PE**12** to the Internet **22-012**. Both EPD's can do local DNS lookups to the Domain Name Services (DNS) server **22-402**.

[0235] Both SRV_AP **22-302** and SRV_AP **300** maintain API communication paths to SRV_CNTRL **200**via **22**PA**02** and **22**PA**08** respectively.

[0236] A Gateway Device (SRV_GW) **22-514** is located in the same region as SRV_AP **22-302** and SRV_AP **300**. This can send emails, process financial transactions and other functionality of SRV_GW devices of the GVN.

[0237] SRV_AP **22-306** connects to the SRV_CNTRL **200** via **22**PA**10** and its egress point to the Internet **22-014** in its region is via **22**PE**20** to POP **22-024** to **22**PE**22** to Internet **22-014**.

[0238] The SRV_GW server **22-516** connects to the SRV_CNTRL **200** via **22**PA**24** and to the Internet **22-014** via **22**PE**26** to POP **22-024** to **22**PE**22** to Internet **22-014**.

[0239] SRV_AP **22-304** connects to the SRV_CNTRL **200**via **22**PA**18** and its egress point to the Internet **22-016** in its region is via **22**PE**26** to POP **22-028** to **22**PA**30** to Internet **22-016**.

[0240] SRV_GW **22-512** connects to SRV_CNTRL via **22**PA**34** and to SRV_AP via **22**PA**16**. Local traffic from SRV_GW **22-516** egresses via **22**PE**28** to POP **22-208** to **22**PA**30** to Internet **22-016**.

[0241] There exist other devices within the GVN and they engage in specific roles such as a backup server SRV_Backup **22-522** and logging server SRV_Logging **22-516**. These are connected to SRV_CNTRL via **22**PA**20** and **22**PA**22** respectively. They can accept data from SRV_CNTR L**200** or from other devices via PA ##paths to be relayed to SRV_Backup **22-522** or SRV_Logging **22-516**.

[0242] This described typology of the GVN allows for traffic from the EPD **100** to have multiple options for its traffic per region through multiple tunnels to multiple SRV_AP servers. The other devices ensure that information is distributed to devices for efficient utilization.

[0243] FIG. **23** illustrates multiple tunnel connectivity between End Point Devices (EPD) **100**, **23-102**, **23-158** and Access Point Servers (SRV_AP) **300**, **23-302**. These tunnels can be used for client data traffic, internal system data or other transfers. This figure further demonstrates the connecting of Global Virtual Network (GVN) infrastructure devices such as Central Server (SRV_CNTRL) **200** and Back Channel Admin Server (SRV_BC) **23-502** with other devices within the GVN.

[0244] The SRV_BC **23-502** establishes and maintains back channel tunnels **23**PA**02** to EPD **100**, **23**P**018** to EPD **23-102**, **23**PA**06** to EPD **23-158**, **23**TP**50** to SRV_AP **23-302** (and on and on).

There may be more SRV_BC servers within the GVN to offer redundancy in the case that one SRV_BC is not operational and also to ensure best performance by placing SRV_BC servers at strategic locations close to devices which they are to connect to.

[0245] EPD **100** connects one LAN **23-002** to various paths which data could take through the GVN such as via one of three multiple tunnels **23**TP**00**, **23**TP**02**, or **23**TP**04** to SRV_AP **300** to an egress point via path **23**PE**00** to Internet **23-410**.

[0246] Another path is from SRV_AP **300** to SRV_AP **23-302** via one of three multiple tunnels **23**TFP**10**, **23**TP**12** or **23**TP**14**.

[0247] A path option from SRV_AP **23-302** is to Internet **23-412** egress point via **23-382**.

[0248] An External Entry Point X-IP **305** from Internet **23-412** into the GVN allows for connectivity by non-GVN devices to address and reach devices through the GVN realizing enhancements of the GVN for the duration of the journey of traffic carried by the GVN.

[0249] Another benefit realized by the GVN is secure tunnel connectivity to an EPD **23-158** at the location of a service providing partner organization in the cloud for secure tunnel via GVN to their servers and related services at location at LAN **23-152**.

[0250] A LAN-WAN-LAN bridge from LAN **23-002** to LAN **23-012** is possible via communication path from **23-002** to **23**CP**02** to GWD **23-004** to **23**CP**04** to EPD **100** to **23**TP**00** **23**TP**02** **23**TP**04** to SRV_AP **300** to **23**TP**10** **23**TP**12** **23**TP**14** to SRV_AP **23-302** to **23**T P**2023**T P**2223**T P**24** to EPD **23-102** to **23**CP**14** to GWD **23-014** to **23**CP**12** to LAN **23-012**. All traffic carried by this bridge is protected and improved by the GVN mechanism.

[0251] Multiple tunnels between two devices such as **23**TP**00** **23**TP**02** **23**TP**04** or **23**TP**10** **23**TP**12** **23**TP**34** or **23**TP**20** **23**TP**22** **23**TP**24** can either offer a single communication path by sending traffic down one tunnel, or two or more tunnels can be aggregated together where two or more bound tunnels can carry traffic as if they are one tunnel.

[0252] SRV_CNTRL **200** with API communication path between peer pairs and tunnels to other devices can be utilized for file transfer and data exchange via paths such as **23**PA**00** to EPD **100**, or **23**TP**30** to **23-302** to **23**FP**22** to EPD **23-102**, or **23**PA**04** to **23-302** to **23**TP**6** to EPD **23-158**, and other potential options.

[0253] There are other possible communication paths in this example embodiments and also more options for communication paths through the GVN. In this example embodiment, all tunnels are representing links via the Third Layer of a GVN with each of them built on the GVN First Layer over top of the Internet.

[0254] FIG. **24** is a simplified example diagram of how the internet works today taking into account hop count or time to live (TTL) as well as path taken due to peering relationships and related routing policies.

[0255] A**0** represents a network of an internet service provider (ISP). A**1** through A**06** represent point of presence (POP) and these POPs further connect to switch devices or client devices to link them to the internet. This hop and spoke structure illustrates clusters of networks within the broader network of an ISP. Lines with circles as line caps denote this connectivity. For simplicity sake, A**1**, A**2**, A**3** and other POP's in this example embodiment do not illustrate links to last mile networks but this should be implied. Each POP has its own hub and spoke connectivity to networks such as local area networks (LANs) or internet data centers (IDCs) which attribute their internet connectivity via the POP.

[0256] H**0** is an example of a single-honed ISP meaning that it relies on one path between it and the internet. If this path is cut or goes down, the connectivity from this ISP to the broader internet is cut.

[0257] B**0** is an example of a multi-honed ISP with five illustrated connections between it and other ISP networks, if one path is unavailable traffic can still flow through the internet although through a less direct route.

[0258] IX**1** and IX**2** are examples of internet exchanges (IX) which may independently link to each other through backbone or backbone dedicated connections. IX s are where ISPs and others can connect to each other at a "meet-me room" or equivalent arrangement for direct network to network peering.

[0259] There are also communication paths between the networks of an ISP and other ISPs or with IX s or with intermediary routers in-between. These backbone communication paths are illustrated by lines with arrow caps on both ends. The intermediary devices are illustrated by circles between the arrow-capped lines. Backhaul connectivity between IX are illustrated by dotted lines with arrow-caps at both ends. An off page connector IBH**1** is used to illustrate International Backhaul (IBH) that the IX**2** also has connectivity to another IX which is not illustrated in this example embodiment.

[0260] To illustrate a direct, efficient connection between ISPs from A**0** to G**0** via the path AX**1-1**->AX**1-2**->IX**1**->GX**1-1** has only four intermediary hops and should be the most efficient route.

[0261] To illustrate a roundabout path caused by the failure of a path, if path GX**1-1** goes down, then traffic from H**0** or A**0** which is destined to transit to G**0** will not be able to go through GX**1-1** via IX**1** The alternative is for traffic to go via B**0** and E**0** to G**0**. What used to take only four middle hops from A**0** of AX**1-1**->AX**1-2**->IX**1**->GX**1-1**, now need many more hops AX**1-1** to AX**1-2** to IX**1** to BX**1-4** to BX**1-3** to BX**1-2** to BX**1-1** to B**0** to EB-**5** to EB-**4** to EB-**3** to EB-**2** to EB-**1** to E**0** to GE-**3** to GE-**2** to GE-**1** for it to reach G**0**. Seventeen middle hops and corresponding higher latency for traffic to now reach G**0** from A**0** if GX**101** is down.

[0262] At the same time, traffic from G**0** to IX**1** which should go through the single middle hop of GX**1-1** will have to go from G**0** to E**0** to B**0** and then to IX**1**.

[0263] This extra traffic could strain connections and be the cause of higher latency and congestion related packet loss. Peering through an IX will usually have much more capacity and ability to handle large volumes of traffic. When the single middle hop GX**1-1** from G**0** to IX**1** is unavailable the added hops (TTL) and latency (RTT) through the alternate route(s) may have too many hops or take too much time resulting in either packets being marked as undeliverable or for internet based services to timeout.

[0264] The best connectivity between two ISP networks via an IX and by utilization of backhaul is represented by the path H**2** to H**0** to HX**1-1** to HX**1-2** to IX**1** to X**1**X**2-1** to X**1**X**2-2** to IX-**2** to DX**2-2** to DX **2-1** to D**0** to D**2** This is 12 hops in total from POP to POP.

[0265] The next direct path would be via B**0** for a total of 16 hops. Path is H**2** to H**0** to HX**1-1** to HX**1-2** to IX**1** to BX**1-4** to BX **1-3** to BX **1-2** to BX **1**-to B**0** to DB-**4** to DB-**3** to DB-**2** to DB-**1** to D**0** to D**2**.

[0266] The next direct path would be via A**0** via C**0** for a total of 19 hops. Path is H**2** to H**0** to HX**1**-to HX**1-2** to IX**1** to AX**1-2** to AX**1-1** to A**0** to AC-**1** to AC-**2** to AC-**3** to AC-**4** to AC-**5** to C**0** to CD-**1** to CD-**2** to CD-**3** to D**0** to D**2**.

[0267] An indirect but possible path could be **30** hops due to routing policies and peering relationships such as via G**9** via E**0** via B**0** via F**0**. Path is H**2** to H**0** to HX**1-1** to HX**1-2** to IX**1** to GX**1-1** to G**0** to GE-**1** to GE-**2** to GE-**3** to E**0** to EB-**1** to EB-**2** to EB-**3** to EB-**4** to EB-**5** to B**0** to FB-**5** to FB-**4** to FB-**3** to FB-**2** to FB-**1** to F**0** to DF-**5** to DF-**4** to DF-**3** to DF-**2** to DF-**1** to D**0** to D**2**.

[0268] Looping occurs when traffic cannot reach a destination because of badly formed or incorrect routing policies governing middle devices between origin and destination. For example, if traffic from C**0** wants to route to G**0** it may choose to go to B**0** thinking that B**0** will send traffic to E**0** because C**0** may think that B**0** and E**0** are close to each other and that this is the best path. However, B**0** may not peer with E**0** directly but have a strong peering relationship with F**0**. F**0** too does not have a peering relationship or a path to reach E**0** and so it may send traffic to DO. DO has only two choices, to send traffic either to C**0** or to B**0**, in both cases the end result is looping, undeliverable traffic. There are other causes of looping such as faulty routing tables, broken devices, hacking or other bad acts or other reasons.

[0269] The net result of too many hops and too high of latency are either time outs or dropped packets.

[0270] FIG. **25** illustrates the strategic positioning of infrastructure to enhance performance. Within this example there exists three or four key points where the strategic positioning of SRV_AP servers and other GVN infrastructure will ensure optimal peering and performance between all points on the example network topology illustrated.

[0271] SRV_AP servers installed and operating at IX**1**-IDC, B**5** and IX**2**-IDC and possibly D**5** for to include optional routing options and failover will provide peering with all other networks and a stable path between SRV_AP with options to route around any broken paths. This strategic positioning offers resiliency and possibilities for other performance enhancements.

[0272] FIG. **26** illustrates how the GVN can incorporate technologies such as Network Slingshot to realize great advantages over distance seamlessly. Network Slingshot is further described in U.S. Provisional Patent 62/266,060.

[0273] The first boundary is GVN EIP **26-322** between the internet and the GVN. The next boundary is the secure perimeter **26-182**. This layered security approach protects the core infrastructure which the GVN is predicated upon.

[0274] The secure perimeter **26-182** boundary between GVN and GVN backbone protect the high speed global network. The section of the GVN above the perimeter **26-822** has traffic flowing over the top (OTT) the open internet via secure GVN tunnels. Under the secure perimeter **26-182**, GVN connections utilize various protocols over dark fiber or other connectivity which are not directly reachable from the internet.

[0275] A super computer node **26-538** can operate inside of (below) the secure perimeter **26-832** which can operate a true internal network with advanced features such as remote direct memory access (RDMA) to a parallel filesystem (PFS) **26-602** device.

[0276] FIG. **27** illustrates how tables on the databases of various GVN devices are related to each other and in which way they interact. For example, the repository database DB_**2300** on the SRV_CNTRL has various tables on it related to devices and the interaction between devices via the neutral API mechanism (NAPIM) of the GVN. Tables such as Device Registry DBT_**2310** in database DB_**2300** is designated as REPO_ACTIVE which means that it receives information from many sources, is read/write and is able to be queried as the source of information for selective or full replication to tables such as Device Identity DBT_**2102** as a part of the database EPD Local Db DB_**2100**. This table DBT_**2101** has the designation SEL_REP+W which allows for selective replication from DBT_**2310** as well as for it to report relevant identity back to the device registry.

[0277] The control and release of information is governed by data managers. Database table type designators include REGULAR for normal, read/write tables, as REP_INFO for read only, replicated tables, as SEL_REP read only, partially replicated tables with only related rows, as REPOS_ACTIVE a table combined from all sources on repository for device registry DBT_**2310** such as identities. Other possibilities include LOGGING from source tables to be combined on the database DB **2800** on SRV_LOGS. These designations for tables are for example only and may be different in real world use and there are many more tables and other types based on use.

[0278] FIG. **28** illustrates the collaborative effort between various modules, mechanisms, technologies and other components of the GVN.

[0279] There are three layers of the GVN layer one is the physical network layer such as the internet on which the GVN is built over the top (OTT) of Layer three is the GVN network layer that client devices see as a partial or complete path to destination. Layer two is the logic layer between the two.

[0280] There are components which interact with the physical conditions **28-00**. Dynamic construct modules at **28-20** strive to maintain connectivity of the GVN. The joint effort section described herein links the relevant modules of the GVN to physical **28-00** and dynamic **28-20** elements. For example, in order for the advanced smart routing (ASR) module G**106** to properly function, there must be multiple access point servers (SRV_AP) GP**106** placed at various locations, with adequate routing and peering GR**106**. In order for an EPD to be able to select the most appropriate SRV_AP to establish a connection with, it needs information about which SRV_AP's are best. The ASR server availability module SA **106** ranks servers for that specific EPD based on information provided by ASR test manager TM **106** and when an EPD requires a new tunnel to be established, it utilizes the server availability list SA **106** in order to build a new tunnel. Tests are then run on that tunnel via TM **106**.

[0281] As another example, for NAPIM G**102** to operate it needs API listeners and handlers HL **102** on a host server. On both host client and host server in the NA PIM, an operations manager OM **302** is running to handle the preparation, then sending, handling, processing of API requests and responses. The dynamic construct of the NAPIM entails peer management PM **102**, management of related NAPIM actions AM **102**, and the transactions at physical TP **102** and dynamic TM **102**.

Construct

[0282] FIG. **29** illustrates the Advanced Smart Routing (ASR) feature of a GVN. Specifically, this figure shows the Advanced Smart Routing (ASR) feature of a GVN within an End Point Device (EPD) **103** to multiple paths to egress points in various regions of the world.

[0283] Traffic in this example embodiment begins in LANA **102** from connected devices such as a Host Client **101** Target regions for traffic illustrated in this example embodiment are: 1) local traffic staying local via POP **401** where performance will not necessarily be improved by a GVN tunnel; 2) local traffic carried in encrypted tunnel TUN**1** to Internet **203**; 3) traffic to another region via TUN**2** to a SRV_AP **301** in that region to access Internet **303**; and 4) Traffic to other remote regions via TUN**3** with some ASR on SRV_AP **501**.

[0284] DNS Cache **103-4** within EPD **103** does DNS Lookups from DNS servers at each target region including DNS **404** for Internet **402**, DNS **204** for Internet **203**, and DNS **304** for Internet **30**M and DNS **504** for Internet **503**. The internal DNS cache **103-4** is accessible via path DP**4**.

[0285] The physical Network Interface Controller (NIC) hardware devices of the EPD **103** includes four ports. ETH**0 103-9** is the WAN port connecting the EPD **103** to the Network Access Point (NAP) to the Internet via P**401** to POP **401** of the ISP on the way to the Internet **402**. All traffic from the EPD goes over this connection as the First Layer of the GVN Network. TUN tunnels on top of this connection are the Third Layer of the GVN. ETH**1 103-1** is the Local Area Network (LAN) port connected to LAN A **102** via path P**102** ETH**2 103-2** is another physical LAN port connected to LAN B **104**via path P**104** Finally there is a virtual interface (VIF) acting as a bridge BR**0 103-3** to bind the LAN interfaces **103-1** and **103-2** via internal paths DP**1** and DP**2** respectively.

[0286] Traffic from LAN bridge BR**0 103-3** is sent to a chain of virtual interfaces (VIF) via device path DP**3**. Advanced Smart Routing (ASR) is applied at each VIF with routing tables of IP Addresses directing flow of traffic to one of two or more exit points from each VIF. The last VIF may have only one possible exit point for “all other” remaining traffic.

[0287] For example, at VIF**0 103-5**, local traffic exits via P**401** All other traffic through VIF**0 103-5** is sent to the next VIF in the chain, VIF**1 103-6** via DP**5**. Traffic from VIF**1 103-6** destined for Internet **203** leaves the EPD **103** via path P**201** through encrypted tunnel TUN**1** to SRV_AP **201** and then to path P**202** to POP **202** to P**203** to Internet **203**. From there, regional DNS lookups can be queried via SRV_DNS **204** via path P**204**. A connection to a Host client **205** or Host server **206** can be made via P**205** and P**206** respectively.

[0288] Any remaining traffic from VIF**1 103-6** is sent to VIF**2 103-7** via path DP**6**. Based on routing tables applied to VIF**2 103-7**, all traffic destined for Internet **303** and connected devices there such as Host server **306** leaves VIF **2**via path P**301** to TUN **2** to SRV_AP **301** and onward through to Internet **303** and beyond.

[0289] Any further remaining traffic from VIF**2 103-7** is sent to VIF**3 103-8**. All traffic from VIF**3 103-8** is sent to SRV_AP **501**via encrypted tunnel TUN**3**. ASR routing is applied at the SRV_AP

501 with traffic destined to IP Addresses within Internet **503** sent via path P**502**, to POP **502** to Internet **503**.

[0290] Traffic from SRV_AP **501** destined for Internet **603** is sent via a connected, encrypted tunnel TUN**4** to SRV_AP **601** to path P**602** to POP **602** to P**603** to Internet **603**, and beyond.

[0291] DNS lookups in the region of Internet **603** can be made to SRV_DNS **604** and connections to devices there can be made for example via P**605** to Host server **605** or other devices.

[0292] This ASR mechanism can be utilized at various traffic junction points for optimizing the sending of traffic to best egress point on the Internet in various target regions, for Geo-Destinating traffic, and other advantages realized by the GVN.

[0293] FIG. **30** illustrates building a series of encrypted tunnels between a Client (C) and a Server (S). The steps **30-0** through **30-18** illustrate a simplified series of communications between C and S.

[0294] The first step is the opening of the connection **30-0** by the C to S. The next step is the acceptance of the connection handshake **30-2** by S. If the handshake data is malformed or otherwise not in an expected form, the process can stop here.

[0295] Upon receipt back and acceptance of the handshake **30-4**, the C presents a certificate to S for it to use along with required security info to build a Secure Sockets Layer (SSL) connection between them **30-8**. This certificate received from C will be compared against the corresponding certificate key on S. If the certificate is expired or incorrect, the SSL connection will not be able to be established and the process will stop.

[0296] This connection will be utilized to send information about the tunnel from the C **30-10** to the S, including pass phrase, metrics and other information concerning tunnel metrics including which IP Address and Port each device will use for tunnel traffic, and other information.

[0297] The S will validate this information **30-12** against its own version of tunnel metrics and pass phrase and other information. If the information is not accurate, the process will halt at this step.

[0298] Upon successful validation, S will send a response back to the C so that it can begin the process of initiating or building the tunnel with the provided configuration settings **30-14**.

[0299] Once the tunnel is up, routes can be applied **30-16** at C or S or both. Although the tunnel is up, during the process of adding routes to it, traffic may not flow through the tunnel or if traffic is able to flow through the tunnel, there exists a risk of data leakage. This risk occurs because until all of the routes have been applied, the traffic to a target IP address may egress the default exit path to the internet without being encrypted or traveling through the tunnel. Once the route has been added to the tunnel, subsequent traffic will be protected as it will be transported through the tunnel. Depending on size of the routing table to be applied to the tunnel, this delay can be a significant amount of time.

[0300] When the routes are all applied to the tunnel, the tunnel is available for traffic to be pushed through it **30-18**.

[0301] FIG. **31** illustrates the flow of information required by two peers in a peer pair. The peers can either be a Client (C) and a Server (S), or a Peer to another Peer in a P-2-P topology. For simplicity of labeling and descriptions within this example embodiment, the C to S and P-2-P represent the same type of two peer relationship, with C to S described herein. The GVN mainly uses C to S relationships between devices but its methods and technology can also be applied to P-2-P peer pairs for tunnel building.

[0302] An encrypted tunnel by its nature is a secure communication path through which data can flow. W hen the Client and the Server are separated by distance and the connection between them is over the open, unencrypted Internet, an encrypted tunnel is an ideal channel through which to safely exchange data. If there is a human network administrator at either end, they can program devices. But there exists a challenge on how to relay security information like pass phrases, keys and other information. Some may use a voice phone call to coordination, others a series of postings through secure web sites to share information or other methods. Manually setting up a single tunnel can be a task. Administering multiple tunnels can become onerous.

[0303] To automatically build a series of encrypted tunnels between two devices in a peer pair there exists a need to securely share information. The tunnel information also needs to be current and securely stored on a device. Furthermore, during the establishment process, there exist threats which must be addressed. While a tunnel is up, other threats exist which need to be addressed.

[0304] SRV_CNTRL **31**D**00** is a central server which contains Repository managing information in Database tables, files stored in a Secure File Storage system, lists in memory and other related information. The SRV_CNTRL also has algorithms and mechanisms to evaluate certain data to generate information reports.

[0305] Client Device **31**D**02** represents a device which will initiate the building of a tunnel by "dialing" the connection to the Server device via a specific IP Address and Port. There can be many Client **32**D**02** devices concurrently connected to the GVN with similar software and configuration with a differentiating factor between Clients of unique device Identity, UUID's, and also unique information per tunnel per Client.

[0306] Server Device **31**D**06** represents a device which will be listening for client connection attempts on specific IP Addresses and Ports. If the Client follows correct protocol and establishment sequence, and presents correct credentials and other security information, the Server will allow the Client to build a tunnel to it. There can be many Server **31**D**06** devices concurrently connected to the GVN with similar software and configuration with a differentiating factor of unique device Identity, UUID's, and unique information.

[0307] Tunnel Info **31**S**2** shows the information stored on a Client Device **31**D**02** and a Server Device **31**D**06**. Each device can establish multiple tunnels and each tunnel will have its own set of Tunnel Information and Security Information. Some tunnel information sets may be used for building of current active tunnels and other tunnel info sets may be held in reserve for future tunnels.

[0308] Certain information between the C and S is equivalent such as a pass-phrase which one will present to the other, and other information will be different depending on applicability. Information requirements for building of a tunnel between two points can include: client/server topology and settings; IP and port of each end point to be used by the tunnel; tunnel metrics including MTU size, protocol, and other information used for its operations; keys, pass phrases and other information about security protections used for the tunnel; SSL certificates and other information for protecting the information exchange pre-tunnel UP; and other information. The information is shared between devices using specific API Action calls of the Neutral API of the GVN.

[0309] Before Tunnel **31**S**0** describes the process of receiving and sharing information between devices **33**D**02** **33**D**06** and the repository **31**D**00** on SRV_CNTRL and back to devices **31**D**02** **31**D**06**. API Communication Paths API-**31**CP**0**, API-**31**CP**2**, API-**31**CP**4**, and API-**31**CP**6** represent Request-Response information exchange with the arrows representing the direction of the flow of information from one device to another device.

[0310] Server **33**D**06** reports information to Receive Info **31**C-**0** module of the SRV_CNTRL **31**D**00** device via path API-**31**CP**0**. SRV_CNTRL **31**D**00** receives information from servers and stores relevant Identity, Tunnel, Current Load and other information in its repository. F or example, algorithms and AI logic on SRV_CNTRL **33**D**00** analyze server load and based on current and anticipated demand from Client **31**D**02** devices, Server Availability C-1 matrix is updated. The Server Availability C-1 information may be conveyed by database replication through the API of the GVN to Clients **31**D**02** via Share Info **31**C-**6** module via API call path API-**31**CP**6**, by direct file sharing via GVN, or other method.

[0311] Client **31**D**02** reports information to Receive Info **31**C-**0** module of the SRV_CNTRL **31**D**00** device via path API-**31**CP**2**. This information will be stored in the repository of SRV_CNTRL **31**D**00**. Specific tunnel information from a Client **31**D**02** can be shared with Server **31**D**04** by Share Info **31**C-**6** module via path API-**31**CP**4**.

[0312] SRV_CNTRL **31**D**00** compiles a current List of Clients **31**C-**4** per server which it publishes to Server **31**D**06** via Share Info **31**C-**6** via path API-**31**CP**4**.

[0313] If either Client **31**D**02** or Server **31**D**06** detects problems in establishment of tunnel utilizing current tunnel information, one or the other device can request a new set of tunnel information to be generated by SRV_CNTRL via API-**31**CP**2** or API-**31**CP**0** respectively. New tunnel info sets can be shared via Share Info **31**C-**6** with both peers in a peer pairing with Client **33**D**02** info sent via API-**31**CP**4** and Server D**02** info sent via API-**31**CP**6**.

[0314] The List of Clients **31**C-**4** and the current state of a Server **31**D**06** will have a direct impact on the Server Availability **31**C-**2**.

[0315] Each Server **31**D**06** needs to organize, secure and coordinate its List of Clients **31**C-**4** which will attempt to build new tunnels to shared resources of Server **31**D**06**. This information will be fluid and need to be updated regularly via secure API calls to SRV_CNTRL **31**D**00**.

[0316] The need to securely harmonize info between devices is essential to protect the integrity of tunnels between them.

[0317] Tunnel Build **31**S**4** phase describes the process of tunnel establishment via Share Info **31**C-**6**. Refer to FIG. **30** for the steps taken between Client and Server to build the tunnel.

[0318] The path **31**TP**0** represents path between Client **31**D**02** and Info Exchange **31**C-**10** and from Info Exchange **31**C-**10** to Server **31**D**06** via path **31**TP**2**.

[0319] Establishment Threats **31**C-**8** refers to threats to Info Exchange **31**C-**10** during tunnel establishment. If the signature of the tunnel type is visible, then there may be threats during the tunnel establishment **31**CC-**8** such as fake Transport Layer Security (TLS) handshakes from illegitimate actors in the middle, TLS errors on handshake, Port and IP identification resulting in blocking or obstructing, time outs due to filtering devices, reset packets sent by ISP or firewall or device in the middle, or other threats.

[0320] If the Info Exchange **31**C-**10** is successful, the Build Tunnel **31**C-**12** step will betaken with routes applied and other related actions to enable the tunnel TUN to be securely built between Client **33**D**02** and Server **31**D**06**.

[0321] Tunnel UP **31**S**6** describes the period during normal flow of traffic through a tunnel. It is essential to convey information between devices and there exists a need on SRV_CNTRL D**00** to manage unique info for various Client **31**D**02** and Server **31**D**06** devices, as well as for multiple tunnels to be built between them.

[0322] The exchange of information between devices has to be a regular occurrence as there exists a recurring need to make fresh, new dynamic tunnels. Some ports on an IP address may be blocked or become blocked and simply changing the port for that IP address will allow the tunnel to be built and for data to flow. Furthermore, each tunnel needs one or more unique ports per IP Address to avoid collisions between tunnels. When a Client **31**D**02** device requests new tunnel information to be created, a random port number is generated and the port availability for that specific IP address on the target Server **31**D**06** is checked against two or more factors including; if that port is already in use by an existing tunnel (either an operational one or one on standby which could be made operational), and if that port has been used by that specific Client **31**D**02**/Server **31**D**06** peer pair in the past and if it has been blocked. In both cases, a new random number will be generated. There are 65,536 ports available per IP address with a certain number reserved for specific services. A floor for example of 5,500 would leave available 60,036 ports which could be used by the random number generator with a min of **5001** and max of 65536. When a tunnel is dismantled and the port is marked as blocked for a peer pair, it is made available to other peer pairs to utilize. This freeing up of ports is necessary to avoid exhaustion of ports. Therefore the tracking IP and Port combinations by SRV_CNTRL **31**D**00** is essential.

[0323] A tunnel can help with its own establishment through steps but it also has limitations. While secure, most tunnels are visible during establishment. The handshake and signature of what kind of tunnel it is may be visible during operation. Manually set keys are cumbersome and not often changed and if used for too long, the risk that they can be broken increases; therefore keys should be re-keyed with new ones on a frequent basis.

[0324] Automated systems need to ensure that information such as new keys, ports to IP Addresses and other info can be created and that this information is available to both sides of the peer pair so that tunnels can be built and rebuilt. Both sides have to be configured & ready to be able to build tunnels. Therefore, the exchange of info between peer pairs needs to be secure or integrity of the security of the tunnel itself is compromised.

[0325] While tunnel is up and pushing traffic, operational threats **31**C-**14** exist. The tunnel signature may be visible (e.g. if the tunnel is sniff-able and not obfuscated). The structure of the tunnel may be known if type of tunnel is able to be discovered. This risks the stream of packets being grabbed and brute force key breaking being used to decrypt the contents of the tunnel. Or a reset signal can break tunnel if the reset code or other tunnel control codes are known. Therefore to maintain tunnel security and integrity between Client **31**D**02** and Server **31**D**06** devices in a peer pair, the updating and sharing of information needs to be automated and secure.

[0326] The GVN structure allows devices to be enabled for automated secure tunnel establishment between peer pairs based on most current information. A combination of security features and methodologies offer self-reinforcing protections.

[0327] FIGS. **32-35** illustrate the third Layer of the GVN with respect to neutrality and security of the GVN tunnel while comparing number of hops to the hops of the base Internet connection. The use of the term LAN in these figures is intentionally general and can represent the network of a home or office or Internet Data Center (IDC). Devices could be clients or servers connected to the LAN. FIG. **32** illustrates a GVN Tunnel from LAN to EPD to SRV_AP to Internet. FIG. **33** illustrates GVN Tunnel from LAN to EPD to SRV_AP to EPD to LAN. FIG. **34** illustrates a GVN Tunnel from LAN to EPD to SRV_AP to SRV_AP to EPD to LAN. FIG. **35** illustrates additional elements, including peering points, of a the GVN Tunnel from LAN to EPD to SRV_AP to SRV_AP to EPD to LAN of FIG. **34**.

[0328] All four figures include the common element of baseline from EH**1** through to EH **17** representing the external hops of the base internet connection. The distance between each hop is not to scale and does not represent anything other than the number of hops. Other common elements include a local area network LAN**1** with a gateway device GWD**1** at one end and another LAN**2** with GWD**2** at the other end. Each variation of this example embodiment also has a GVN end-point device EPD-**1** connected to an access point server AP-**1**. There exists a tunnel between these devices and one neutral hop per device NH**1** and NH**2** inside of the Third Layer of the GVN.

[0329] FIG. **32** illustrates a GVN Tunnel from LAN to EPD to SRV_AP to Internet. The tunnel can also act in the other direction offering entry access from Internet to GVN tunnel back to LAN. There is a point of presence POP-**1** between AP-**1** and the Internet. Another POP-**2** is between the Internet and GWD**2** representing a Network Access Point (NAP) for connectivity of that LAN.

[0330] FIG. **33** illustrates GVN Tunnel from LAN to EPD to SRV_AP to EPD to LAN. This variation illustrates an end-to-end GVN tunnel(s) between edges of two LAN s via one SRV_AP. The difference between this variation and FIG. **32** is that the tunnel extends over the entirety of the transit through the Internet from EH**3** through EH**15**. A second EPD-**2** is illustrated.

[0331] There is one tunnel between EPD-**1** and AP-**1**. This is joined to a second tunnel between AP-**1** and EPD-**2**. There are three neutral hops within the Third Layer of the GVN represented by NH**1**, NH**2**, and NH**3** as compared to the 13 hops on the base Internet between EH**3** and EH**15**.

[0332] Total hop count from LAN**1** to LAN**2** is therefore at minimum seven hops from LAN**1** to GWD**1** to NH**1** to NH**2** to NH**3** to GWD**2** to LAN**2** The end to end count includes two internal hops at both ends from EH**1** through EH**17** and totals a minimum of 17 hops.

[0333] FIG. **34** illustrates a GVN Tunnel from LAN to EPD to SRV_AP to SRV_AP to EPD to LAN. This variation illustrates an end-to-end GVN tunnel(s) between edges of two LANs via two (or possibly more) SRV_APs. The difference between this variation and FIG. **33** is that there is a second AP-**2** inserted into the path to represent another joining of tunnels A P-**1** to AP-**2** and AP-**2** to EPD-**2**. There is another internal neutral hop added brings the hop count inside the Third Layer of the GVN to eight.

[0334] FIG. **35** illustrates additional elements, including peering points peering points between ISPs and network edges, of a the GVN Tunnel from LAN to EPD to SRV_AP to SRV_AP to EPD to LAN of FIG. **34**. This variation illustrates an end-to-end GVN tunnel(s) between edges of two LAN s via two SRV_APs and it also further illustrates more information about different Internet Service Providers (ISP) carrying traffic over certain portions of the Internet between EH-**3** and EH-**15**.

[0335] The difference between this variation and FIG. **34** is that additional elements have been indicated. The following elements as illustrated in FIG. **9** have been overlaid in this variation of the example embodiment: a) EDGE-**1** is the demarcation point for network access connection between the devices of LAN-**1** and the POP of ISP-**1**; b) PP-**01** is the point where peering occurs between the ISP-land ISP-**2** networks; c) PP-**02** is the point where peering occurs between the networks of ISP-**2** and ISP-**3**; and d) EDGE-**2** is the demarcation point for network access connection between devices of LAN-**2** and the POP of ISP-**3**.

[0336] Some advantages can be realized by placing SRV_AP-**1** at PP-**1** so that this SRV_AP directly can peer with both ISP-**1** and ISP-**2**. More advantages can be realized by placing SRV_AP-**2** at PP-**2** so that this SRV_AP can directly peer with both ISP-**2** and ISP-**3**. If the network of ISP-**2** is not ideal, it is possible for traffic to be alternatively routed around ISP-**2** by the GVN through another route or line or ISP or carrier.

[0337] The hop count through the neutral Third Layer of the GVN remains at eight as it was in FIG. **34**. The distance between ISPs is not to scale. Furthermore, it is likely that there could be more hops within the network of an ISP but for simplicity sake, the quantity illustrated has been simplified.

[0338] While FIGS. **33**, **34** and **35** all illustrate the joining of tunnels at AP hops, this is viewed as a single tunnel by client devices within LAN**1** and LAN**2**. This singular tunnel represents the neutral Third Layer of the GVN within which it is possible to run all traffic that would normally transit over the internet, including TCP, UDP, and other protocols, plus other tunnels such as IPSec, OpenVPN, PPTP, or other. There are other advantages realized by the Third Layer of the GVN. Some include lower TTL and ability to have more control over routing plus other advantages.

[0339] FIG. **35** illustrates the weaving together of various network fabrics into a network tapestry. This example embodiment illustrates the weaving together of various network fabrics at the physical layer over the top (OTT) of which the global virtual network (GVN) operates. These fabrics at the physical layer **36102** constitute a chain of network segments which may for example be IPv4 and IPv6 aware, or only one or the other protocol. The end point device (EPD) **36100** to LAN (**36000**) could be IPv4 and/or IPv6. The tunnel TUN **36**P**2** can be one or the other or both protocols between the EPD **36100** and the access point server (SRV_AP) **36300**.

[0340] The Egress/Ingress point (EIP) **36302** denotes the exit and entry point from the GVN to network fabrics at the internet level. The path **36**P**04** denotes the connectivity to IPv4 internet networks **36400** and the path **36**P**06** denotes connectivity to IPv6 internet networks **36600**. The key point is that the Tapestry of the GVN allows for end to end links of fabrics such as IPv4 internet **36408** to IPv4 in the LAN **3600** or end to end IPv6 **3608** from Internet **36600** to LAN **36000** even though there may be some dissimilar segments at the physical level **36102**.

[0341] FIG. **37** illustrates communication pathways in a GVN for automated device collaboration. This example embodiment shows the communication pathways such as P**37202**-C used by a neutral API mechanism (NAPIM) API **37202** **37206** **37208** for automated interaction between various devices working together to constitute a global virtual network (GVN).

[0342] Key operational aspects can be automated to facilitate rapid systemic response. These include infrastructure operations, heartbeat routines, connectivity, testing & diagnostics, and other functionality.

[0343] Infrastructure operations such as keeping device operating system software and packages up to date from reliable sources with predictability, maintaining GVN modules and databases and other operations. For example, an end point device (EPD) **100** can query the central control server (SRV_CNTRL) **200** via API **37202** along paths P**37202**-B to **37202**-C. In another example, the email gateway server (SRV_GW_Email) **37310** can update a system package from SRV_CNTRL **200** which is a reliable source for trusted system software.

[0344] Other items such as heartbeat functionality running via daemons or other repeat cyclical operations include keeping services up, running and healthy with reporting from a device such as an access point server (SRV_AP) **300** reporting to SRV_CNTRL **200** via API **37202** through paths P**37202**-A to P**37202**-C. There are also exist redundancy paths such as via API **37208** through paths P**37208**-A to P**37208**-C. Other heartbeat functionality can keep queues operating and clear, can replicate logging data and other such operations.

[0345] For connectivity such as tunnels P**37206**-C between EPD **100** and SRV_AP **300** relevant information is required by both ends of the tunnel, the listener at SRV_AP **300** and the initiator EPD **100**. This information can include peer pair info related to each device or related to the tunnels. Both communicate with SRV_CNTRL **200** via independent paths via API **37202**.

[0346] With multiple tunnels hooked on to virtual interfaces and the option for more than one tunnel between devices such as EPD **100** to SRV_AP **200** or SRV_AP **200** to SRV_AP **20***x*, various different API calls are required for the management of multiple tunnels, routes and other information.

[0347] The algorithms which power server availability rely on systemic analysis of various kinds of information to offer EPDs with a list of SRV_AP servers with which they can connect via tunnels. Since each tunnel requires an IP address and port at either end mapped to the GVN construct so that routing is clear, is changing information needs to be up to date. Automated device collaboration facilitates this.

[0348] A key component for information sharing is for that of testing & diagnostics data from the Layer 1 physical network, from the GVN construct Layer 3 as well as from logic at the GVN Layer 2. This connectivity information provides more information for analysis on the SRV_CNTRL **200**. Replication of this data can also be to a logging server via API **37208** or other communications path. The results of the analysis can also be stored on the logging server.

[0349] The API can also be utilized to update information about itself to each peer in a pairing such as peer pair credentials, ID, and other info, the queue on each, the transaction logs for reconciliation, by internal security audits and for updating or adding or deprecating action functions of the API mechanism itself.

[0350] Systems and resources monitoring and reporting is also key with information conveyed automatically about services up and running, that hosting is working, that the database engine is up and running, that security systems are running and more.

[0351] FIG. **38** illustrates the problems and challenges of dynamic tunnel building. This example uses the transfer of files, database structures and other data from a Repository **38**R-**00** to a Device **38**D-**00** of the GVN to illustrate the problems and challenges of dynamic tunnel building. The Repository **38**R-**00** will in most cases be on the Central Server (SRV_CNTRL) of the GVN. Device **38**D-**00** can be an End Point Device (EPD), Access Point Server (SRV_AP), Gateway Server (SRV_GW_XX), or other device of the GVN.

[0352] Depending on the type of device, a newly created device may be loaded with a clone of a master disk to be configured during first boot or as in the case with remote servers, a first boot script will be securely transferred to the server to be run to pull base system files. Other potential scenarios may combine a combination of pre-loaded files plus files to be loaded remotely.

[0353] Upon running of a first boot script, most current database structure is replicated from Repository **38**R-**00** from DB Structure Repository **38**R-**06**-A to Db **38**D-**04** on Device **38**D-**00**.

Data to populate the database will be send from DB Data Repository **38**R-**06**-B via **38**P**06** to an Identity Info module **38**S-**00**. Some data passing through **38**S-**00** may be filtered and modified to incorporate identity information such as Device ID and other UUID elements and other data passed through as direct copy without modification.

[0354] Depending on the type of device and the device's universally unique identifier (UUID), data appropriate for device **38**D-**00** is sent via path **38**P**16** to Database **38**D-**04**. Some information may also be populated into template configuration files which can be cloned to the Software and Configuration Files storage **38**D-**02** on device **38**D-**00**. Identity information unique to a device may include: device attributes, naming and UUID info, credentials/keys, key adjustors, other information.

[0355] Settings files for system packages and other modules will be cloned from Settings Files Repository **38**R-**02**-B on Repository **38**R-**00** and sent via path **38**P**02** to Software and Configuration Files storage **38**D-**02** on device **38**D-**00**. Some “factory default settings” and other files may also be copied via path **38**P**10** to Secure File Storage **38**D**06** on the device **38**D-**0**. The Secure File Storage **38**D-**06** is administered by the Files and Folder Manager of the GVN. Files from **38**D-**06** may also be cloned to **38**D-**02** via **38**P**12** when needed, such as in the case of having to revert back to factory settings.

[0356] Code Base Files **39**R-**02**-A from Repository **39**R-**00** can be copied to Software and Configuration Files storage **38**D-**02**via path **38**P**00** and also can be copied to Secure File Storage **38**D-**06** via path **38**P**8**.

[0357] The above illustrates the loading of files and data from repository to device during first boot, updates, regular data exchange, and other operations.

[0358] FIG. **39** illustrates the bridging of two LANs into a wide area network (WAN) via two or more EPDs. More particularly, this figure shows the bridging of two LANs **39-000** and **39-010** into a wide area network (WAN) via the EPDs. Each EPD first connects to an access point server SRV_AP **39-200** via base tunnels built over the top (OTT) of their internet connections.

[0359] From EPD **39-100**, the base connectivity path OTT is via paths **39**PO**02** to a point of presence (POP) **39-002** to the internet **39-004** to the POP **39-006** of the SRV_AP **39-200**. From EPD **39-110**, the base connectivity path OTT is via paths **39**-PO**12** to a point of presence (POP) **39-012** to the internet **39-014** to the POP **39-016** of the SRV_AP **39-200**.

[0360] The transit path **39**-P**06** from POP **39-006** to POP **39-016** could be the path through the internet, by passing the SRV_AP and relying on the routing on the public network. If the EPD **39-100** wants to connect to EPD **39-102** via the internet, it may follow a different route based on policies out of the control of the GVN or either EPD.

[0361] EPD **39-100** builds a tunnel TUN **39**-P**10** between itself and SRV_AP **39-200** EPD **39-102** also builds a tunnel TUN **39**-P**12** between itself and SRV_AP **39-200**. One or both of these tunnels may or may not be encrypted or secured. There can also be another tunnel, internal tunnel INT TUN **39**-P**20** run through both of the other tunnels, joined at the SRV_AP **39-200** through which traffic can flow. This tunnel can be the communications path through which the WAN is built.

[0362] The tunnel and base connection connectivity can use different network protocols. Network tapestry offered by the GVN can be a blend of different network protocols mapped to a chain of various network segments while concurrently the GVN can be one network type end-to-end within the internal tunnel.

[0363] FIG. **40** illustrates a Multi Perimeter Mechanism (MPFWM) running on a GVN. This example demonstrates how there can be a second degree **40**TOP**88** over the top (OTT) element within a global virtual network (GVN). At the first degree of OTT **40**TOP**86**, the GVN **40-86** operates OTT the base internet connectivity **40-82**. In the case of a multi-perimeter firewall mechanism **40-88** construct, it is operated OTT the GVN and so therefore can be construed as a second degree over the top element **40**TOP**88**.

[0364] FIG. **4** illustrates a GVN stack build over the top (OTT) of the Internet. This example describes the GVN **41-800** stack built over the top of the internet **41-000**. The figure shows the connectivity between EPD **100** and two SRV_AP servers **300** and **302** via tunnels TUN **41-100-300** and TUN **41-100-302**. These two tunnels are an example of multiple tunnel options between EPD and the best current access point server (SRV_AP) based on server availability and other factors such as destination, type of traffic, QoS of various network segments between origin and destination, and more.

[0365] Tapestry **41-500** is the weaving together of various network protocols of individual network segments as well as the end to end protocols which can be "run through" GVN paths.

[0366] Cluster GVN Devices **41-600** represents the physical layer of routes between devices of the GVN.

[0367] GVN Global Network OTT Internet+via other Links **41-700** is the GVN Layer 2 logic where modules operate such as geodestination, DNS management, advanced smart routing (ASR)/global ASR (GASR), server availability, tunnel management and builder module, etc.

[0368] GVN **41-800** represents the network that the client user sees.

[0369] FIG. **42** compares the internet protocol IP stack B**2**, the OSI model C**2**, and the GVN stack C**3**.

[0370] The IP stack consists of Network Interface T**1**, Internet T**2**, Transport T**3**, and Application T**4**.

[0371] For non-GVN traffic and for the physical tunnel invisible to the client egressing through ETH NIC N1, the IP stack seen by the client follows elements R**1** at the Network Interface T**1** layer, R**2**A at the Internet T**2** layer, R**3**A or R**3**B at the Transport T**3** layer, and R**4**A, R**4**B or R**4**C at the Application T**4** layer.

[0372] For traffic through the GVN tunnel and network a client will view its GVN traffic at R**4**C at the Network Interface T**1** layer, R**5** at the Internet T**2** layer, R**6**A or R**6**B at the Transport T**3** layer, and R**7**A, R**7**B or R**7**C at the Application T**4** layer.

[0373] While the OSI model may be used by clients for IP traffic through the tunnel, the GVN has its own stack of Network Interface G**1**, Internet G**2**, Transport G**3**, GVN routing & logic G**4**, GVN Internet G**5**, GVN Transport G**6**, and Application G**7**.

Logic

[0374] FIG. **43** illustrates global Internet flows between countries via many possible routes. Traffic on the global Internet flows between countries via many possible routes transiting different interconnects between peers.

[0375] Internets of countries within regions, such as Asia, are mainly connected to each other both by ground and submerged oceanic links. Typically they are chained where traffic from one country to another transits a third or more countries in the middle.

[0376] **43**-X**01** represents the most direct route from Asia to Europe. For example from Hong Kong to Paris via **43**-X**01** latency will be between 180 ms and 250 ms depending on route taken. **43**-X**02** is an indirect, longer path where traffic is naturally pushed through the Internet.

[0377] Traffic here goes from Asia to West Coast USA **43-400** via link **43**-P**400** then to East Coast USA **43-402** via link **43**P**402** and then to landing point in Europe **43-600** via link **43**P**600**. The latency via **43**-X**02** will be approximately 396 ms to 550 ms or more depending on destination within Europe.

[0378] Prior to leaving a region, traffic may have to relay from one country to one or more other country(ies) before it can access the International Backbone. For example, traffic from China **43-000** may travel through link **43**P**002** to Hong Kong **43-002** and then to Japan **43-006** via link **43**P**006**. These extra in-region hops can add 50 ms to 150 ms or more to an RTT even before traffic leaves a region.

[0379] Once in the destination region, traffic will land in one country for example in the UK **43-600** from trans-Atlantic link **43**P **600**. From the UK **43-600**, traffic will travel via link **43-600** to France **43-602** and then on to Germany **43-606** via link **43**P **606**. These extra in-region hops can add 30 ms to many more ms to the RTT depending on destination.

[0380] Quality of International Backhaul also can vary between peers with each having various RTT QoS times. The routes and corresponding speeds on the normal Internet are decided middle-men actors and these are in the majority of cases based on least expense often delivering slow RTT speeds. High latency is not the only issue to contend with on the lower quality networks. These usually have higher levels of congestion and correspondingly high packet loss. Lossy and slow links significantly degrade performance.

[0381] FIG. **44** again compares the internet protocol IP stack, the OSI model, and the GVN network stack. This example again compares various conceptual network models such as the TCP/IP stack B**2**, the Open Systems Interconnection model (OSI) A**2**C**2**, and also variations such as the TC P/IP model within the GVN stack A**3**, as well as the model of the GVN C**3**.

[0382] There are two perspectives presented. The Client Perspective A**1** compares A**2** and A**3** side by side. The Global Virtual Model framework C**1** compares C**2** with C**3**. There also exists a tree connecting layers of B**2**.

[0383] Within the TCP/IP model B**2**, there is the Network Interface T**1** corresponding with Ethernet Protocol R**1** Internet T**2** corresponds with the Internet Protocol (IP) R**2** The Transport T**3** layer corresponds with the TC P protocol R**3**A and the UDP protocol R**38**. Other protocols may exist and operate at this layer. On top of this is the Application Layer T**4** where the Hypertext Transfer Protocol HTTP R**4**A, mail service POP**3**R**4**B, and GVN applications reside. Other applications such as file transfer protocol (FTP) or other services can reside within this layer.

[0384] To compare the TCP/IP model with the OSI model in the scope of B**2**, the OSI Data Link S**9** and Physical Link S**8** are parallel with T**1** The OSI Network S**10** is parallel with T**2** The OSI Transport S**11** is parallel with T**3**.

[0385] The OSI Session S**12**, Presentation S**13**, and Application S**14** layers are within the scope of R**4**C, the GVN application.

[0386] The TCP/IP model through the GVN B**3** builds an extension to the network tree atop of R**4**C.

[0387] From the Client's perspective, the layers T**1**, T**2**, T**3**, T**4** combine into a single TCP/IP model layer T**5**, becoming a Network Interface Layer for the neutral Third Layer of the GVN. T his compares with the OSI Model A**2** Physical S**1** and Data Link S**2** layers.

[0388] On top of R**4**C, there exists representations of the internal layers within the Third Layer. The internal IP Layer is at R**5** and this corresponds with the A**3** level of Internet T**6** and the A**2** Network level S**3**.

[0389] TCP protocol R**6**A and UDP protocol R**6**B and this level corresponds with A**3** level Transport T**7** and A**2** level Transport S**4**. Other protocols may exist and operate at this layer.

[0390] The Application layer from the Client's Perspective T**8** corresponds with internet protocols such as FTP R**7**A, HTTP R**7**B, and POP**3**. The OSI model breaks the Application layer T**8** into three layers, Session S**5**, Presentation S**6**, and Application S**7**.

[0391] Within the GVN's three layers model, A**1** describes operations within the Third Layer while B**1**, B**2** describes operations within the First Layer. The GVN application R**4**C at T**4** and the operations under C**1** describe how the Second Layer functions to allow the Third Layer to operate on top of the First Layer.

[0392] There are similarities between the operations of the network in the Third Layer and First Layer of the GVN.

[0393] The Network Connectivity NO can be other over the regular Internet N**1**, via a WAN N**2**, Dedicated Circuit N**3**, MPLS Line N**4** or other link to the Internet.

[0394] FIG. **45** illustrates an tunnel between two LANs via the GVN. Specifically, this figure describes the internal path from LAN **45-000** to LAN **45-002** through the GVN path **45**P**00** to **45**P**10** the segments through the internal tunnel **45**L **300**. There are five hops **45**H**0** through **45**H**8** visible to the client in either direction between the two LAN s. The path through **45**L **300** is the G

V N Layer visible to clients.

[0395] The GVN Level 1 Network Layer **45**L **100** represents the physical network layer for various different types of network segments end-to-end. While not demonstrated within this figure the number of hops and network segments is at least equal to and most likely greater than those visible to the clients within the internal tunnel **45**L **300**.

[0396] The logic layer Level 2 Logic **45**L **200** is the logic where various network segment integration, routing and other GVN operations occurs.

[0397] If the client path is IPv6 through the tunnel, for IPv4 segments only like **45-104**, the internal IPv6 traffic can be encapsulated in such a manner so that it can remain native IPv6 end-to-end regardless of the network type at the network layer **45**L **100**.

[0398] FIG. **46** compares the network at the base level via paths P**01** through P**13** to the network through the GVN T**01** through T**03**.

[0399] Significant measurements at the base internet level CTN **140** are LAN to GVN via EPD **46-100** to SRV_AP **46-300** for which connectivity metrics for bandwidth BW, latency Δt=A ms, Packet Loss, and other factors are evaluated. At the other end of the connection, similar measurements BW, Δt=C ms, Packet Loss and other factors at CTN **142** measure the on-ramping of traffic into the GVN from EPD **46-102**. Through the GVN between SRV_AP **46-300** and SRV_AP **46-302** for the GVN trans-regional OTT various internet segments CTN **340** measure BW, Δt=B ms, Packet Loss, and other factors are evaluated. Overall path latency through the GVN Layer Three GVN**4-3** can be calculated as the sum of the latencies of A+B+C for total in milliseconds.

[0400] At GVN Layer Three GVN**4-3**, ASR and other features govern how and where traffic flows through the GVN. This entails determining the best tunnel to send traffic through based on target region and traffic type, QoS of the segments through the GVN and other factors.

[0401] At GVN Layer One GVN **4-1**, the physical conditions of the base network connectivity are monitored and tested to determine best route options on top of which to build GVN tunnels and pathways through them. GVN pathways can transit through joined tunnels passing through SRV_AP, SRV_BBX and other GVN hardware devices. This can also determine which tunnels to make, to continue using and which to deprecate.

[0402] Mechanisms, modules and component parts at GVN Layer Two GVN **4-2** help to set up, test, manage and otherwise operate the plumbing between Layer Three GVN **4-3** and GVN Layer One GVN**4-1**. Tunnel testing **46-310** can be done in Layer Three at the EPD **4100** and at the SRV_AP **46-300** via its tunnel tester **46-312**.

[0403] FIG. **47** illustrates the Advanced Smart Routing (ASR) feature and elements of the Geo-Destination Mechanism of a GVN within an End Point Device (EPD). This includes using multiple DNS Sources to send traffic via multiple paths to egress points in various regions of the world. Target regions for traffic illustrated in this example embodiment are: 1) local traffic stays local from VIF**3 47-118** to Internet **47-004**; 2) traffic destined to other region Internet **47-002** will go from VIF**1 47-112** through TUN**1 102-6** to path **47**P**48** to SRV_AP **47-300** and then on to Internet **47-002** via path **47**P**50**; 3) traffic for other region Internet **47-006** will go from VIF**247-116** through TUN**2 102-8** to path **47**P**52** to SRV_AP **47-302** and then on to Internet **47-006** via path **47**P**54**; and 4) traffic for other region Internet **47-008** will go from VIF**3 47-118** through TUN**3 102-10** to path **47**P**56** to SRV_AP **47-304** and then on to Internet **47-008** via path **47**P**62**.

[0404] SRV_AP **47-304** includes more detail to illustrate functionality some of its components AP Logic **47-314** and Content Pulling Agent **47-318**. In addition, the EPD **100** includes a flow chart of more detail to illustrate its internal functional components.

[0405] The tunnels TUN**1 102-6**, TUN**2 102-8**, TUN**3 102-10**, and, Traffic Flow through VIFs with routes table applied at each of virtual interfaces VIF**1 47-112**, VIF**2 47-116**, VIF**3 47-118** operate in a similar manner to virtual interfaces and tunnels.

[0406] The DNS Cache **47-114** is seeded from multiple DNS Sources both via Local DNS Query mechanism **47-100** through path **47**P**38** to Internet **47-004** to SRV_DNS **47-104** via **47**P**34**. The

Remote DNS Query mechanism **47-108** can make DNS queries via Content Pulling Agent (CPA) **47-318** via **47**P**44** to SRV_DNS **47-114**.

[0407] The Geo-Destination Mechanism (Geo-D) pushes routing information to Routing Manager **47-104** via **47**P**04** connecting Content Delivery Agent (CDA) **47-106** with CPA **47-318**. The path **47**P**30** to **47**P**40** via J**01** is an abstraction to represent the coordination of CPA **47-318** working with CDA **47-106**. Communication between CPA & CDA is still via tunnel and or API call, or transfer via chained caches, through tunnels, or can be via other mechanism.

[0408] In this example embodiment, through Geo-D, the CPA **47-318** pulls all regional content into SRV_AP **47-304** via **47**P**62** from Internet **47-008** to pull content via **47**P**66** from Host Server **47-110** that is hosting destination content, and within which content the CPA **47-318** may discover links for other content and the CPA **47-318** will then pull a content stream from Host Server **47-108** via **47**P**64**. Other content may be pulled from Host Server **47-112** via **47**P**68**. It is typical for many web sites to host pages on one server, video files to stream from another server and for graphics to be served from another server.

[0409] FIG. **48** illustrates examples of various concurrent types of paths for traffic to take via the GVN. The left side of EDGE-**1** represents the LAN side. The right side represents the Internet facing side. The right side of EDGE-**2** represents the LAN side and the left side represents the Internet facing side.

[0410] Traffic from devices within LAN **001** leave the EPD **101** via P**002** through an encrypted tunnel P**003** to SRV_AP **102** and can egress to the general Internet **106** to reach Host Client or Server devices D**005** via path H**005**. Traffic from devices within LAN **201** leave EPD **301** via P**103** to SRV_AP **302** and can egress via P**106** to the Internet **106** to reach Host Client or Server devices D**005** via path H**005**.

[0411] EPD **101** can link to EPD **301** via the Internet **106** through P**003** to SRV_AP **102** to P**006** to Internet **106** to P**106** to SRV_AP **302** to P**103** to EPD **301**. There are secure tunnels between the EPDs and SRV_APs via paths P**003** and P**103** To ensure complete security, for end to end secure tunnels, the path between EPDS is EPD **101** to P**005** to SRV_AP **103** to P**007** to WAN **107** to P**107** to SRV_AP **302** to P**105** to EPD **301**.

[0412] EPD **101** can build a secure tunnel to SRV_AP **102** via P**003** and from there link to another secure tunnel via P**201** to WAN **103** to P**202** to SRV_AP **104** and then egress to the Internet **105** in a remote region via path P**203** and on to Host Client or Server devices D**002** via path H**002**.

[0413] EPD **301** can build a secure tunnel to SRV_AP **302** via P**103** and from there link to another secure tunnel via P**301** to WAN **303** to P**302** to SRV_AP **304** and then egress to the Internet **305** in a remote region via path P**303** and on to Host Client or Server devices D**004** via path H**004**.

[0414] EPD **101** is also able to reach devices in Internet **305** via secure tunnels between EPD **101** to SRV_**102** to SRV_AP **302** to SRV_AP **304** and egressing to the Internet **305** from there.

[0415] EPD **301** is also able to reach devices in Internet **105** via secure tunnels between EPD **301** to SRV_**302** to SRV_AP **102** to SRV_AP **104** and egressing to the Internet **105** from there.

[0416] There are many other options for routing via end-to-end tunnels, tunnels to egress points on the open Internet, tunnels via multiple SRV_AP devices and other options.

[0417] A n important point illustrated by this example is that client traffic which is carried by the GVN is through a GVN Third Layer which from the client perspective is the same as a path through the Internet and is consequently able to carry any type of traffic through it while still realizing the benefits of the improvements and greater degree of security offered by the GVN.

[0418] For example, path P**008** illustrates WAN Optimization connectivity between a firewall GW**002** device and a firewall GW**202** device to create a LAN-WAN-LAN bridge. The device to device communication is carried inside Third Layer of GVN and is transparent to GW**002** and GW**202**.

[0419] For simplification purposes, point of presence (POP) network access points have not been illustrated in this figure. A path to or from the Internet such as Internet **105** to device D**002** does have a POP in the middle of H**002**.

[0420] The WAN in this example embodiment represents a secure tunnel between GVN devices on top of the Internet, and so any mention of WAN is at the Third Layer of the GVN with all GVN traffic still transiting the First Layer.

[0421] FIG. **49** describes the automated advanced smart routing (ASR) from one device at the start **49-000** to the end device **49-800**. If a route is not available, automated advanced smart routing can built a route, including but not limited building new tunnels, and updating of internal routing for most optimal path.

[0422] Tables 1 through 5 are utilized by this algorithm as data points to use for routing purposes, such as determining best egress point for traffic from GVN through access point server to the open internet. This data can also be used by the algorithm to help prioritize which route is better with respect to another route.

[0423] Table 1 lists various available paths from origin to destination and include a rating for path ranking.

TABLE-US-00001 TABLE #1 Rate the QoS of various routes through a GVN

RT_ID	Path from Origin to Destination	Rating
1	EPD custom-character .Math.POP.Math.Internet.Math.Destination	0.15
2	EPD.Math.TUN1.Math.SRV_AP1 custom-character .Math.POP.Math.Internet.Math.Destination	0.36
3	EPD.Math.TUN2.Math.SRV_AP2 custom-character .Math.POP.Math.Internet.Math.Destination	0.58
4	EPD.Math.TUN2.Math.SRV_AP2.Math.SRV_AP3 custom-character .Math.POP.Math.Internet.Math.Destination	0.96
5	EPD.Math.TUN3.Math.SRV_AP2.Math.WAN.Math.SRV_AP4 custom-character .Math.POP.Math.Internet.Math.Destination	0.85

[0424] EPD custom-character and SRV_AP2custom-character denote denote Egress/Ingress Points (EIP) from a device to/from the internet. The two sided arrow ⇄ symbol indicates the routed path between two devices. This can either be directly through the internet, as a network segment OTT the internet as a tunnel or other mechanism (possibly as part of a GVN) or via other network path between devices. The point of origin is on the left and the destination implies the final location where the traffic is to be routed to/from.

[0425] The rating is a calculated value for a route based on a number of factors. A rating of 0.00 implies an impossible route. A rate of 1.00 implies the most perfect route with highest bandwidth at wire-line speed latency. RT_ID is the route ID number to differential one route from another both for utility, testing and logging purposes. This is utilized to determine the quality of various routes through the GVN. RT_ID is an identification for a specific route from a list of routes.

[0426] Table 2 describes a server availability matrix.

TABLE-US-00002 TABLE #2 Server Availability Matrix

SA_ID	Server_ID	IP_Addr_ID	Port	PRI	EPD_ID	Param	Flag_State	Timestamp
1	8	236	3581	99	1	[array]	1	1448674236
2	8	235	19501	98	1	[array]	0	1448674237
3	7	218	36152	55	2	[array]	0	1448674237
4	5	158	25739	80	1	[array]	−1	1448674238
5	19	1672	59081	75	2	[array]	1	1448674238

[0427] The information held in the Server Availability Matrix include the Server_ID, the server IP_Address_ID, the port number, the EPD_ID field, a parameter field (including security and configuration settings, a flag state, and a timestamp.

[0428] The PRI is the weighted priority order of servers for the EPD to use to connect with. A priority of 1 is the absolutely lowest priority. 0 indicates that a server is currently not reachable. This is differentiated in the Flag_State which indicates if the record is current or not. PRI may be kept in same table or in another related table as this is a continuously changing value and another table will allow for historical logging and analysis.

[0429] A Flag_State of 0 indicates that it is a standby entry. Flag_State of 1 indicates that it is active and that it can be used. Flag_State of −1 indicates that it is retired, not to be used.

[0430] Table 3 illustrates latency for a complete path as well as latency for component network segments.

TABLE-US-00003 TABLE #3 Route −> Path latency evaluations LAN EPD GVN Egress Total .Math. .Math. Transport .Math. Latency RT_ID EPD SRV_AP To Egress Destination for Path Flag_State Timestamp 1 1 ms — — 236 ms 237 ms 1 1448674236 2 1 ms 18 ms 169 ms 12 ms 200 ms 1 1448674237 3 2 ms 23 ms 135 ms 22 ms 182 ms 1 1448674237 4 1 ms 21 ms 139 ms 8 ms 169 ms 1 1448674238 5 1 ms 21 ms 135 ms 19 ms 176 ms 1 1448674238

[0431] Paths from a LAN via EPD through GVN and or internet or combination of various network segments have a Total Latency for the Path, otherwise known as the RTT, the round trip time. This is the time in milliseconds (ms) for an ICMP ping from origin to destination and its return back to origin.

[0432] In order to evaluate best route, it can be broken down into groups of network segments which make up constituent parts of the total network path. Evaluation of various segments can provide information about routing and offers a data point which can be used. Path rating will always give extra priority weighting to traffic to transit GVN OTT of the internet versus traffic transiting the open internet.

[0433] The latency for the total path is the sum of the latencies: LAN to EPD plus EPD to SRV_AP plus GVN Transport plus GVN Egress to Destination.

[0434] Table 4 lists the measured quality of service attributes of a route.

TABLE-US-00004 TABLE #4 Route −> QoS factors measured (current and historical) Other L_ID RT_ID Reg_ID Load SEC RTT R BW EFF factors Flag_State Timestamp 0 1 1 1.5 1.0 1.1 1.00 2.00 1.22 [array] 1 1448674236 1 6 1 0.8 1.0 0.9 0.97 0.90 0.80 [array] 1 1448674237 2 4 86 1.0 1.0 1.0 1.00 1.00 1.00 [array] 1 1448674237 3 5 44 0.7 1.0 0.3 0.50 0.45 0.75 [array] 0 1448674238 4 7 49 0.25 0.8 0.8 0.90 0.30 0.95 [array] 0 1448674238 5 5 44 0.9 1.0 0.9 0.8 0.78 0.90 [array] 1 1448848558 6 9 44 1.0 1.0 1.0 1.0 1.1 1.1 [array] 1 1448848559

[0435] This table is kept as a log of current and historical QoS (quality of service) results for a route between a source peer and another peer in another location and or region. It can be used in real-time to make QoS expectative decisions based on real-world conditions. This table is located on each origin device and indicates the performance of routes.

[0436] Various factors are used to evaluate for line quality comparisons. These include System load (Load), Security (SEC), Latency (RTT), Packet Loss (R—reliability), Bandwidth (BW), Hop Count (EFF—efficiency), and Other factors (an array of values which can be used to evaluate line parameters).

[0437] A baseline for each point and the network segments between them is utilized so that comparisons can be made between resources which have different hardware configurations and network speed, bandwidth, and other ratings.

[0438] L_ID indicates the ID of the row for the logged route information.

[0439] RT_ID is the path id. The path could indicate a path through the base internet, through a tunnel, joined tunnels, or other GVN related routes.

[0440] Reg_ID is the target region ID.

[0441] RTT is Round-trip-time or latency based on the historical norm. A value of 1.0 is normal while greater than 1.0 indicates a lower than usual latency and lower than 1.0 indicates higher than usual latency.

[0442] SEC is security rating. A value of 1.0 is secure, and a value of 0.0 indicates completely insecure and a totally compromised resource. This is based on security tests, performance logging and other data points. Any value lower than 1.0 is cause for concern.

[0443] R is reliability and is related to Packet Loss on a route. For example R=0.97 indicates 3% of packet loss on a route. A value of R=1.0 indicates 0% package loss and 100% reliability. A rating greater than one indicates parallel duplication of packets sent down a route. R=2.0 indicates 100% reliability for duplicate packets sent.

[0444] EFF indicates the efficiency of the line with respect to the length of the route in terms of hop count and is based on its historical mean. An EFF value of 1.0 implies normal hop count and less than 1 implies greater than usual hop count. A value greater than one implies a lower than usual hop count.

[0445] BW (bandwidth) is based on line ratings fora base connection combined with the complete network segment between two points. A value for BW of 1.0 implies that 100% of the BW is available. A value of 0.5 implies that only 50% of BW is available based on route BW rating. And if a value is greater than one, such as 2.0 then this implies that 200% of that route's BW capacity rating is available and can be utilized. For example, for a 1 GigE base connection between two points, a rating of 0.55 indicates that 550 M bps is available. A rating of 2.0 indicates that 2 GigE can be utilized, etc.

[0446] In the case of RT_ID=1, the SEC (security) value of 1.0 indicates that it is 100% secure, and the greater than one values of RTT=1.1 and BW=2.0 indicate that connectivity of that route RT_ID from one point to another point has a 10% lower latency and double the bandwidth of a comparable, baseline performance of an average route between those points.

[0447] For example where RT_ID=5, the security rating of 0.80 indicates that there is an ongoing security risk, and correlated with the available BW rating of 0.30 shows that the server is under an attack such as DDoS or Brute Force where multiple security threats such as an onslaught of multiple concurrent requests which saturate the available BW (bandwidth) while degrading SEC (security).

[0448] Flag_State=1 indicates current, active route. And Flag_State=0 indicates historical route performance no longer in use. Timestamp indicates when in time as a UN IX timestamp (seconds since the epoch).

[0449] L_ID=3 and L_ID=5 demonstrate the comparison between route from origin to region Reg_ID=44 at two different UNIX Timestamps of 1448674238 and 1448848558. It shows that the performance at the later time has improved since the rating at the prior time. Load is better at Load=0.9 versus Load=0.7 and network connectivity fundamentals have also improved.

[0450] This table can also be utilized to determine the better of two routes from origin device to the target region by comparing the QoS factors of each route. For example L_ID=5 and L_ID=6 both indicate current (Flag_State=1) routes from origin to Reg_ID=44, although the routes are different with RT_ID=5 and RT_ID=9. The better of the two across the board is RT ID=9 and would be weighted with a higher priority in a server availability list.

[0451] Table 5 evaluates and ranks list of egress ingress points (EIP) in target regions.

TABLE-US-00005 TABLE #5 EIP in regions

EIP_ID	Reg_ID	QoS	BW	Load	S_ID	IP_ID	ATR	Flag_State	Timestamp
1	1	0.5	1.0	0.38	25	60	[array]	1	1448674258
2	1	6.1	10.0	0.21	39	128	[array]	2	1448674258
3	2	0.68	1.0	0.33	50	1851	[array]	1	1448674259
4	2	0.14	0.2	0.91	54	1938	[array]	−1	1448674270
5	2	0.72	1.0	0.12	68	2188	[array]	1	1448674272

[0452] The ATR field is the attribute field. This is an array of attributes used to describe the specification for the EIP (RA M, Cores, Storage space, other factors, etc.). The S_ID field holds the Server ID. The IP_ID field holds the IP address ID. Bandwidth (BW) is measured in GigE. For example 20 M bps is 0.02, 100 M bps is 0.1 and 1 GigE is 1, and 40 GigE is 40.

[0453] QoS (quality of service) represents the current EIP (egress ingress point) suitability for a server to handle connections and traffic. A QoS of 1.0 represents the ideal state of a server for an EPD to connect to with acceptable available BW (bandwidth), and little to no Load (resources load of the server, a combination of RAM, CPU, NIC and other factors).

[0454] A QoS of less than 1.0 means that that server is being utilized. If a QoS approaches zero, then that means that it is close to total uselessness due to saturation of capacity. A s a benchmark and for the health of the system, QoS of less than 0.40 will indicate that a server will be prioritized with a much lower rating so that servers with healthier QoS are weighted to appear higher on the list and therefore will attract connections and to not overload any current server.

[0455] This evaluation and ranking mechanism can also be used as a determining factor on how to scale out the build of the physical infrastructure.

[0456] FIG. **50** illustrates the Secure Perimeter **50-182** between the BB/Backbone layer Below Perimeter **50-832** and the IP/Internet layer Above Perimeter **50-822**.

[0457] There are two natural protections in place. The first is the only way to join the two layers together is via paths **50**-TR**6**B**22** and **50**-TR**6**B**32** and that must pass through the secure perimeter. Only valid GVN traffic can transit in either direction via both logical checks. The other safeguard in place is that the network types above and below the secure perimeter **50-182** are different.

[0458] FIG. **51** is a flowchart of Advanced Smart Routing (ASR) within a Global Virtual Network (GVN).

[0459] From the starting point of a host client **101** device in a local area network (LAN) **102** connected to an end point device (EPD) **103**, the GVN offers the EPD a multitude of connection paths to multiple potential termination points. This is flowchart is a high level view of the routing logic a packet could take as it transits a GVN utilizing ASR for optimal performance. From the perspective of the host client **101**, their traffic will flow through an internet protocol (I P) network with as few number of hops and best possible latency at the third layer of the GVN. The first layer of the GVN is the base internet with automatic configuration of a construct of virtual interfaces, tunnels, routing and other networking policies. The second layer of the GVN is where the algorithms, software and logic to govern operation between layer three and layer one.

[0460] The first main routing decision is at a logic gate **104** within the EPOD where traffic either egresses to the local Internet **107** where the EPD is located via path P**104** or if it is to go through a secure wrapped and obfuscated tunnel via P**107** to the access point server (SRV_AP) **110** offering the best connectivity to the region where SRV_AP **110** is located. Prior to traffic egressing SRV_AP A**110**, it passes through a routing logic gate **111**. Traffic to egress locally to the internet **113** will go via path P**111** to either a host client **115** or a host server **116** there. If traffic is not local but rather to be relayed to another region, it will go via path P**116** through a tunnel **118** to the next SRV_AP **119**.

[0461] At SRV_AP **119**, three of many possible routing options are illustrated by the paths that traffic can take. There is a logic gate **126** to determine if traffic should remain and egress to the local internet **129** or if it should go through a tunnel via P**126** to a SRV_AP in another region **127**. Another possibility is illustrated via path P**119** which demonstrates a tunnel from SRV_AP **119** to another EPD **121** in a distant region. This is an EPD **103** to EPD **121** bridged via multiple bridged tunnels.

[0462] A further possibility is for traffic to reach client devices **125-126** in the LAN **122** where EPD **121** is located through the EPD's connection P**121**.

[0463] FIG. **52** is a flow chart of the various routes available through a GVN from an origin C**52-002** to destination S **52-502**. There can be many more possible combinations that are not shown or discussed.

[0464] Path **52**P**00** from the Client C**52-002** to the EPD **52-108** can be used to measure the performance from the client through the LAN to the EPD. Matching of best routes is achieved after tests and evaluating real-time data of available paths. GVN ingress from EPD via first hop **52**P**00** to an access point server (SRV_AP) **52-102**, **52-104**, **52-106**, **52-202**, **52-204**.

[0465] Paths from EPD to first SRV_AP can be defined as the ingress point from the EPD into the GVN and measured accordingly. Internal hops from SRV_AP to SRV_AP follow internal routes which always try to maintain the best path connectivity. These could be OTT internet, over backbone, over dark fiber, or other related routing.

[0466] Best egress points out of the GVN are also kept track of locally, in that remote region and also holistically for the entire network segment from origin to destination.

[0467] Tests can be run on each segment, combinations of segments, and the total network path from end to end taking into account various factors to evaluate. Traffic type and path determination can be depending on data attributes and profile QoS requirements. The main path choice is always based on best factors for traffic over path. A function of this mechanism is to match paths between destination and origin to flow for best possible bidirectional route.

[0468] Table 6 is a list of IP addresses to keep local based on IP Address, protocol(s), and port(s).

TABLE-US-00006 TABLE #6 IP Addresses to keep local LRI_ID IP4_Address Protocol Ports ATR Flag_State Timestamp 1 36.12.22.88 * * [array] 1 1448674102 2 204.68.207.18 TCP 80, 443 [array] 1 1448674103 3 38.12.251.82 TCP, 22 [array] 1 1448674258 SCTP, UDP 4 66.220.12.150 UDP 554 [array] 0 1448674360 5 8.8.8.8 TCP, 953 [array] 1448674361 UDP

[0469] This table keeps track of which IP Addresses to keep local so that transit an EIP (egress/ingress point) either directly on the EPD or via an SRV_AP in the same region as the EPD.

[0470] The LRI_ID field holds the Local Route IP Address ID. A Region value of 0 indicates IP Address(es) to keep local that should go directly to the Internet from the EPD from its local EIP. Region values of 1 to 300 indicate countries and territories. Higher Region ID's represent more finely grained granularity. The IP**4**_Address field holds an IPv4 address.

[0471] Under a column such as Protocol or Ports, an asterisk (“*”) means a wild card covering all potential values in an allowed range or in a list set of allowed values. If one or more values is in a column and separated by a comma, then it indicates that more than one port, or protocol, or other column value can be used. Then only those values explicitly noted will be effected by the table prescription, with other values which are not specified as following the default behavior.

[0472] Table 7 is a list of IP address ranges, their target geographic destination ID, and which EPD ID that these rules apply to.

TABLE-US-00007 TABLE #7 Tables of IP Addresses to route via Geographic Destination GDR_ID IP4_Start IP4_End GDReg_ID EIP_ID ATR Flag_State Timestamp 1 8.4.4.2 8.4.4.8 1 1025 [array] 1 1448674102 2 201.1.2.5 201.1.2.25 1 1026 [array] 1 1448674103 3 38.12.251.82 null 3 3025 [array] 1 1448674258 4 66.220.12.76 66.220.12.80 1 1025 [array] 1 1448674360 5 151.8.11.1 151.8.15.255 5 5093 [array] 1 1448674361

[0473] The GD Reg_ID field holds a geographic destination ID. A Region value of 0 indicates IP Address(es) to keep local that should go directly to the Internet from the EPD from its local EIP. Region values of 1 to 300 indicate countries and territories. Higher Region ID's represent more finely grained granularity. The IP4_Start and IP4_End fields hold starting and ending IPv4 addresses.

[0474] Table 8 is a base reference of IP addresses for countries and other geographic regions to be utilized by a geographic destination mechanism. Due to the large number of IP addresses utilized, the CIDR notation is utilized.

TABLE-US-00008 TABLE #8 Base reference of IP Addresses per region CIPB_ID GDReg_ID Region CIDR4 Total_IP4 Flag_State Timestamp 1 1 US 3.0.0.0/8 16,777,216 1 1448674102 2 1 US 5.1.94.0/24 256 1 1448674103 3 1 US 5.10.70.0/29 8 1 1448674258 4 44 UK 2.22.128.0/20 4096 1 1448674360 5 49 DE 2.16.6.0/23 512 1 1448674361

[0475] This table defines the IP Address ranges for country wide blocks or regional blocks, depending on granularity utilized for regional routing. Geographic destination routed addresses are routed in order before regional IP address tables and therefore route first.

[0476] The CIPB_ID field holds a Country IP Address Block ID. The CIDR4 column indicates the CIDR for a range of IPv4 addresses. CIDR stands for Classless Inter Domain Routing which is a notation to describe a range of IP addresses. For example a slash eight (/8) notation represents a block of 16.78 million IP addresses. A slash twenty (/20) represents 4,096 IP Addresses. The Total_IP4 column indicates the total number of IPv4 addresses which are covered by a CIDR4 defined range.

[0477] FIG. **53** is a flowchart of an algorithm governing the selection of traffic routing from a Start device to an End device.

[0478] In the GVN, there are routing tables for pathways at the base level of the internet between GVN devices and each other such as an EPD and an SRV_AP over top of which a tunnel can be built. Routing tables govern both the Level 1 (internet level) traffic as well as routes through the GVN at Level 3. At times, tunnels may not exist or if they do, they may not be optimal. GVN routing can be mapped both to existing and possible GVN routes based on the topological database. Full information about base network segments, and links between devices is stored in the GVN databases.

[0479] The algorithm starts by identifying the target region for the particular GVN traffic. Next, a check is made to see whether or not a path exists through the GVN **5306**. If this does not exist, a new tunnel **5310** is built. The next step is to check the health of the tunnel **5312**. If it is not okay, a new alternative tunnel will be built **5310**. Once a healthy tunnel is available, routing health is checked **5320**.

[0480] If a path to EIP in the target region **5322** exists route and the route is checked to see if it is the best route for the traffic type. If it is, the route is used **5360**.

[0481] If the route is not ideal for the data type, a check to see if an alternative does exist **5350**. If it does, then the best route for traffic type is utilized **5352** and that best route is used **5360**. When a route is used, processes evaluate the route performance **5365**. Before the algorithm completes, another process will save performance data in logs related to server availability via P**5328**, to list of EIPs **5322** and also to mapping paths to target regions used by **5302**.

[0482] If the test at **5350** determines that the route is not ideal for data type and no alternatives exist, then new tunnel will be built **5310** via path P**5314**.

Control

[0483] FIG. **54** illustrates the modules required for automated device collaboration and information exchange in a GVN.

[0484] EPD **100** is the endpoint device. SRV_AP **300** is an access point server which is located in the target destination region. SRV_CNTRL **200** is a central server accessible by both the EPD and the SRV_AP as well as by other devices which may support a graphic destination mechanism.

[0485] Each device EPD **100**, SRV_AP **200** and SRV_CNTRL **300** stores information about itself in a local information repository in the form of lists, files, database tables and records, and other means. This repository also contains information about peer device relationships, stores logs, plus other relevant operational information. The SRV_CNTRL **200** also has additional storage functionality and its role is to provide information to other devices relevant to them and or to the peer devices which they may connect with, to evaluate current conditions and provide centralized control-like guidance such as the publishing of a server availability list and other functionality. A neutral API mechanism (NAPIM) can send info between devices and the peers which they connect with, and can also be used to update the API itself.

[0486] The database on the SRV_CNTRL **200** acts as a repository for information about itself as well as a centralized repository for other devices. There can be many different SRV_CNTRL **200** servers acting as multiple-masters in many locations. Each database can store certain information including tunnel information, peer information, traffic information, cache information, and other information. Security and other aspects are independently managed by each device including heartbeat functionality, triggered scripts and other mechanisms.

FIG. **55**

[0487] FIG. **55** illustrates the communications between EPD **100**, SRV_CNTRL **200**, and SRV_AP **300** via the neutral API mechanism (NAPIM) of the GVN via paths API-**55**A**1**-**55**A**2**, API-**55**A**3**-**55**A**2** and API-**55**A**1**-**55**A**3**.

[0488] For tunnels TUN **55-1**, TUN **55-2** and TUN **55-3** to be built between EPD **100** and SRV_AP **300** as well as for tunnels from EPD **100** to other SRV_AP servers such as TUN **55-4** and from other EPDs to SRV_AP **300** via TUN **55-5**, each device in the peer pair requires certain information per tunnel.

[0489] The NA PIM mechanism stores relevant credentials, coordinates and other information for each side of a peer pair to utilize when building new tunnels via the Tunnel Managers **55110** and

55330. The server availability mechanism **55222** on the SRV_CNTRL **300** evaluates the performance of various tunnels tested on the EPD side via Tunnel Tester **55112** and the SRV_AP side by Tunnel Tester **55312**. The information from the tests is relayed to the Connectivity Analyzer **55288** on the SRV_CNTRL **200**. Test results include assigned IP address and port combinations, ports used, results from historical combinations use, results from port spectrum tests, and other related information.

[0490] Server availability lists present the EPD **100** with a list of IP addresses and ports which could be utilized by the Tunnel Manager to build new tunnels. The SRV_AP **300** and other SRV_AP servers noted on the list will be notified to expect **55320** and to listen for connection attempts to be made by EPD **100**.

[0491] Server availability prioritizes the list of SRV_AP IP address and port combinations based on expected best performance of the tunnels to be built while also looking at current load of available SRV_AP servers, balancing assigned lists given to other EPDs as well as other available information.

[0492] FIG. **56** illustrates various types of communications available between GVN devices via the NAPIM.

[0493] Closed loops are available as NA PIM REQ/RESP communications between known peer pairs and there are two main types; Device to Repository **56**-P**2**C and Device to Device **56**-P**2**P.

[0494] RESTful URL posting is an open access (if allowed for that specific action) possibly to unknown peers, such as generic or general non sensitive information which can be shared.

[0495] Each defined API action has a flag governing access via path type with potential values, another flag regarding whether authentication is required or not required, plus other controls. For example, an EPD **100** can request a list of available servers and corresponding IP addresses and ports via **56**REQ**100200** and receive that list from SRV_CNTRL **200** via response path **5**RESP**100200**. At the same time, the SRV_AP **300** may be informed by the EPD **100** via **56**REQ**100300** or may receive the information from the SRV_CNTRL **200** either via NAPIM, by data base replication, via back channel, or other message.

[0496] FIG. **57** describes API call groups **57202**, **57206**, and **57208** between different types of devices within a Global Virtual Network (GVN). Each API call is circular in nature with a request sent from a client to a server with a response sent back. In most cases, the client can be one or the other end of a peer pair as long as the other peer has listening enabled for it to act as a server.

[0497] API call group **57202** represents calls from the Central Server (SRV_CNTRL) **200** via path P**57202**-C to End Point Device (EPD) **100** via P**57202**-B and Access Point Server (SRV_AP) **300** via P**57202**-A. This type of communication can exchange information between the repository database and file store on the SRV_CNTRL **200** and the EPD **100** and SRV_AP **300** about tunnel info, logging info, billing info, device peer pair data, and other forms of relevant information.

[0498] Between the EPD **100** and SRV_AP **300** are two types of communication path. The direct tunnel is where Third Layer traffic, information and binary files can be pushed as data packets via path P**57206**-C. There also exists an API call framework **57206** between EPD **100** and SRV_AP **300** via P**57206**-B to **57206** to P**57206**-A.

[0499] Direct connect between EPD **100** and SRV_AP **300** via API **57206** can be for information sharing, collaboration and verification and, other information. For example, an attempt to restart a tunnel can usually be initiated by one side with the other side automatically responding and rebuilding it. However, in the case where a tunnel is stuck and cannot be rebuilt, the API can be used to send commands to try to force a tunnel restart on both ends and if still unsuccessful can share information between devices. This information may trigger a need to use new tunnel information to build a different tunnel between the two devices, or to have both devices query SVR_CNTRL **200** to obtain fresh tunnel building info. Having a communication path between them via API **57206** is therefore extremely useful.

[0500] API call group **57208** represents calls from the CNTRL_SRV **200** and internal backend infrastructure devices and other infrastructure supporting devices of the GVN via path P**57208**-C. For simplicity sake of the illustration, some gateway devices are illustrated in this example embodiment and there are other types of infrastructure devices in a GVN not illustrated here which could connect via this path to the SRV_CNTRL.

[0501] SRV_GW_Email **57310** represents an email server and is linked to CNTRL_SRV **100** via P**57208**-B**1** to **57208** to P**57208**-C. The email can be sent and received via Email Network Access Point (NAP) **57401**. A dedicated email server allows other devices to be focused in their functionality and also offers simplified administration as it is the only device type that needs to be maintained with respect to email server administration.

[0502] SRV_GW_FIN **57318** represents a financial gateway server through which credit card and other financial related transactions could be made to third parties via External API **57501** NAP. Like the example of the SRV_GW_Email, a single focused device type role allows for other devices to focus on their core functionality and presents a simplified administration with only SRV_GW_FIN servers requiring extra administration to protect financial transactions with third parties.

[0503] SRV_GW_Other **57315** represents other types of gateways between the GVN and other services on the internet. Communications between these types of gateway servers and SRV_CNTRL **200** is via P**57208**-B**3** to **5728** to P**57208**-C.

[0504] A secondary API path between SRV_AP **300** and SRV_CNTRL **200** is via P**57208**-A to **57208** to P**57208**-C and exists for redundancy purposes and infrastructure related communication between this peer pair.

[0505] Another group of calls from SRV_AP servers allow a path from SRV_AP **300** to SRV_GW_Email **57310** via path P**57209**A to **5720** to P**5720**-B**1**, and to SRV_GW_FIN **57218** via P**57209**A to **57208** to P**57209**-B**2**, and to SRV_GW_Other **57315** via P**57208**-A to **57208** to P**5720**-B**3**. These can be for API calls for data exchange directly from SRV_AP **300** to those devices.

[0506] The API calls which transit via P**57208**-A can also represent relayed API calls from other devices via the SRV_AP **300** such as from an EPD **100** to SRV_GW_FIN **57318** via path P**57206**-B to **57206** to P**57206**-A to **300** to P**57208**-A to **57208** to P**57208**-B**2** where the flow of the API call through SRV_AP **300** is only another hop in the chain with client being one end EPD **100** and server being the other end SRV_GW_FIN **57318**.

[0507] API calls and other types of information exchange are essential to the operations of devices in the GVN. There are number of type of automated infrastructure operations. These include keeping device operating systems configuration up to date, updating software packages of the O/S and modules from reliable sources to repository which can house update software for ease and predictability of patching, updating and new installs, deploying new global virtual network software modules and keeping installed modules up to date, controlled replication of the GVN database(s), keeping the API actions library up-to-date, and other operations.

[0508] On each device, there are daemons and heartbeat functionality where automation and device to device interaction is required. This includes keeping daemons running, keeping services up, keeping queues up and keeping them unclogged, heartbeat functions, logging functions.

[0509] The connectivity and construct structure in the GVN includes virtual interfaces (VIFs), tunnels, multiple tunnels, routes, server availability, geo-destination, DNS, and caches and chained caches.

[0510] The most up to date information is required for tunnel building and this information needs to be shared between client and server or tunnels will not be able to be built. A s such, there exists a need for testing and diagnostics with the reporting of results data to be analyzed centrally in order to have visibility of the overall operations of the GVN. The testing and diagnostic information can include first layer conditions, connectivity of the tunnels, best point to point route on the Internet, Advanced Smart Routing for best route through the GVN, and device operational status.

[0511] The API can also be utilized to convey information about itself such as peer pair information, queue information, transaction logs, security/accounting and other logs, and API actions, patterns, data structures, and related scripts to process actions either on client or server.

[0512] Information regarding the state and configuration of Hosting Services can also be conveyed via API calls from a device to SRV_CNTRL or other devices. This information can include services up/down status, API module up/down status and if answerable, hosting status of sites, database status, Secure Socket Layer (SSL) certificates status, GVN component status (e.g. whether components such as geo-destination are running).

[0513] There exists other uses for information exchange via API related to Security/FW/Monitoring/Collaboration/Information Exchange, and other mission critical aspects of the GVN. The API is a powerful medium for information exchange and is a holistic self-healing mechanism is therefore possible to be deployed across devices.

[0514] FIG. **58** describes the steps taken for an API call from initiation on a client device Peer (Source) **006** through to sending to server device **007007**B with return back to client **006006**B.

[0515] The API transaction is triggered at API Start **001**. Data is passed to a common class or other type of handler to Create Inner Payload **002**. It is added to a queue **003** which can be in memory, saved to database, flat file or other mechanism. The queue step may be bypassed with API call immediately sending or can be set to send at a certain time. As a part of the heart beat functionality of the client device **006** and depending on the priority flag of the API call in the queue, the payload can be processed immediately, processed at a specific time or deferred based on a factor such as load, queue **003** length, network conditions or other factors. When an item is processed from the queue, an outer payload is prepared and relevant transaction data **004** are generated for a specific, single use API call. W hen the outer API REQUEST payload is ready to be sent it is conveyed via the Neutral API mechanism **005** to be sent to the Peer Target **007** Host (server) API through the Internet Q**01** via path CP**01** to Q**01** to CP**03** or through a secure tunnel WAN Q**02** via path CP**02** to Q**02** to CP**04**.

[0516] Upon receiving **008** the Request payload RP**01**, the server **007** will then begin to parse and interpret the payload. In the processing of the request payload RP**01** there will be security and data integrity checks made and the outer payload will be decrypted to discover the contents of the inner payload **009**. Further security and data integrity checks will be made comparing the inner and outer payloads. Upon validation, the payload is passed to the corresponding script to take the prescribed action **010**. Upon completion of requested action, the inner payload for the response is created **011**. The outer payload creation **012** and transaction preparation **013** follow the same process to create the outer API RESPONSE payload RP**02** as was followed for when the API Request outer payload RP**01** was created. The response is then sent back via Neutral API **014**.

[0517] The API RESP (Response) RP**02** follows the same path back from API server **007** to API client **006**.

[0518] The API RESP RP**02** is received back **015** by the Peer Source API client device **006**. The payload is parsed **016** and processed **017**. Based on API action type, the data received back will be passed to API handler scripts on **006**. Transactions are all Logged **018**.

[0519] If Call Back **019** is specified **020** then a new call will be initiated via path P**019** and in parallel via path P**0** the original API call terminates at API Complete **022**.

[0520] If Call back is not specified **021** in the API RESP RP**02** then the original call proceeds to termination point **022** via P**021** to complete the transaction.

[0521] FIG. **59** is a flowchart outlining the interaction between the EPD and SRV_AP to achieve geographic destination functionality. Specifically, this figure describes the process flow of the geographic destination mechanism starting at client **000** and following a sequential and at times parallel communications path from CP**0** to step **12** end point device (EPD **100**) with EPD **100** interacting with access point server (SRV_AP **300**).

[0522] This process flow ends when the content has been pulled within the remote region to SRV_AP **300** and then sent via transfer and caching within the geographic destination mechanism back to the EPD **100** to be served at step number **15** back to the client **000** via path CP**203**.

[0523] Content pull at step number **8** are in parallel via CP**13**, CP**14**, CP**12** from content servers SRV **803**, **804**, **802** and the results are sent back via CP**10** for listing and then processing of data pull.

[0524] Steps **3**, **12**, **13** and **15** occur in the origin region with respect to the client **000** and the EPD **100**.

[0525] Steps **2**, **1 11**, and **14** are steps that occur in transit in either or both directions between the EPD **100** and the SRV_AP **300**.

[0526] Steps **5**, **6**, and **9** occur on the SRV_AP **300**.

[0527] Steps **3**, **4**, **7**, and **8** occur via EIP (egress/ingress point) from the SRV_AP **300** over the internet in the remote region in which it is located.

[0528] Step **3** is for the DNS lookup for the initial URL, URI, and URN of the content requested by the client **000**. Step **7** is for DNS lookups for nested content to be pulled as constituent parts of the initially pulled content.

[0529] FIG. **60** describes device collaboration within a geographic destination as an overview with component parts indicated as modules and their constituent parts noted on the various devices including the information stored in memory and databases and information exchange and communicated via communication paths both for API traffic as well as data transfer such as file transfer between devices. A GVN makes it possible to control complex automated structures spread across multiple devices to work together to achieve a shared goal.

[0530] The figure shows the components of EPD **100** and illustrates the geographic destination mechanism on an end point device (EPD). The figure also shows the components of SRV_AP **300** and illustrates the geographic destination mechanisms on an access point server (SRV_AP **300**) in a remote region from the EPD.

[0531] The content pulling agent D**302** resides on the SRV_AP **300**. The CPA D**302** receives the target URL/URI from the CDA D**102** located on the EPD. This target address that the client wishes to reach is located in another region from the client and is where the client wishes to pull content from. The CPA D**302** passes the request address to the remote fetcher bot (R.F.BOT **301**).

[0532] R.F.BOT D**301**'s job is to do the DNS lookup D**304** and then to use that information to pull content via data pull **301**. The R.F.BOT D**301** works in conjunction with CPA D**302** to parse the fetched results via CP**01** to seek any other addresses for auxiliary content which can and should be pulled as constituent parts of that content. Requests are stored in database D**302** for access and future reference by CPA D**302** and R.F.BOT D**301** The content file list L**301** is passed from R.F.BOT D**301** to CPA D**32** Data files content is passed from DATA PULL **301** via R.F.BOT D**301** to Cache Manager D**303** Pulled files are sent to the cache manager D**303** for on transfer either as a file clump or as separate files.

[0533] Depending on distance from origin to geographic destination region, the file type and QoS, the pulled files in the cache may be clumped into one single file for unified transfer through the chained cache or as individual files which may be sent in parallel, concurrent streams.

[0534] There are multiple optional paths to remote region. Data can be transferred via paths between API and TP**01** to TP**02**, between TP**01** and TP**03**, and between TP**02** and TP**03**. Data files can also be transferred through the GVN via paths CP**38**, CP**39**, or P**06** to CPBB, etc. CP**38** is a path via tunnel from SRV_AP **300** to SRV_AP D**555** via GVN D**888**. CPBB is a backbone path between SRV_AP D**555** and SRV_AP **300** via relay SRV_AP D**505** path P**06**. CP**39** is the file transfer path over GVN from cache **701** to EPD **100** via SRV_AP D**555** CP**02** indicates a direct connection path possibility between SRV_AP **300** and EPD **100**.

[0535] The optional path to remote regions allow options for traffic to flow via best route based on current conditions, network segment attributes and how these contribute to best transfer, data type, as well as other factors.

[0536] FIG. **61** illustrates how a globally distributed parallel file system (PFS) can be connected via a GVN. Specifically this figure illustrates how a globally distributed parallel file system (PFS) can allow access to one of three **61308**, or **61318**, or **61328** PFS storage nodes seamlessly using native RDMA access through a GVN Tapestry over the top (OTT) of various non-native network fabrics to realize the required quality of service (QoS) and adhering to the high performance computing (HPC) principles required for this functionality.

[0537] PFS **61308** is an example of one PFS instance in a client's LAN behind an EPD linked to two other PFS instances “in the cloud” with native RDMA over IB between all three PFS storage nodes allowing true parallel access regardless of network type at base segments. The link **61**CP**06** is the base internet connection between EPD **100** and SRV_AP **300** and TUN**1** runs OTT of **61**CP**06**. **61**CP**10** is either within an IDC or OTT internet. PFS **61308** connects to PFS **61318** via paths **61**CP**08**->**8**P**02**->**8**CP**06**/TUN**1**->**8**CP**10**->**8**CP**32**->**8**CP**18** which represents short distance within a region. These devices are both located within the same High Performance Zone.

[0538] SRV_AP **300** connects to SRV_BBX **61310**via **61**CP**10** and both are located within the same Global Node.

[0539] PFS **61318** connects to PFS **61328** via SRV_BBX **61310** connecting to SRV_BBX **61320** which represents Global Node to Global Node long distance communication via GVN.

[0540] The present disclosure is not to be limited in scope by the specific embodiments described herein. Indeed, other various embodiments of and modifications to the present disclosure, in addition to those described herein, will be apparent to those of ordinary skill in the art from the foregoing description and accompanying drawings. Thus, such other embodiments and modifications are intended to fall within the scope of the present disclosure. Further, although the present disclosure has been described herein in the context of at least one particular implementation in at least one particular environment for at least one particular purpose, those of ordinary skill in the art will recognize that its usefulness is not limited thereto and that the present disclosure may be beneficially implemented in any number of environments for any number of purposes. Accordingly, the claims set forth below should be construed in view of the full breadth and spirit of the present disclosure as described herein.

Claims

1. A method comprising: identifying, by one or more processors, a plurality of candidate communication paths between a first network node and a second network node; determining, by the one or more processors, a plurality of security ratings corresponding to the plurality of candidate communication paths, wherein the plurality of security ratings indicate a level of security risk for each of the plurality of candidate communication paths; and selecting, by the one or more processors, the communication path from the plurality of candidate communication paths based on the plurality of security ratings, wherein the selected communication path is used to communicate data between the first network node and the second network node.