What is claimed:

1. A method for managing a plurality of instances of a Wide Area Network (WAN) optimizer executing on an intermediary device, the method comprising:

(a) establishing, on a device intermediary to a plurality of clients and a plurality of servers, a plurality of instances of a Wide Area Network (WAN) optimizer to accelerate WAN communications between the plurality of clients and the plurality of servers;

(b) monitoring, by the device, network traffic traversing the device for each of the plurality of instances of the WAN optimizer; and

(c) selecting, by a manager executing on the device responsive to the monitoring, a change of a load balancing scheme to load balance the plurality of instances of the WAN optimizer.

2. The method of claim 1, wherein step (a) further comprises automatically establishing, by the device, a configuration of a size of each of the plurality of instances of the WAN optimizer based on data stored from monitoring of previous execution of the plurality of instances of the WAN optimizer.

3. The method of claim 1, wherein step (a) further comprises executing, by the device, each of the plurality of instances of the WAN optimizer as a virtual machine in a virtualized environment.

4. The method of claim 1, wherein step (b) further comprises monitoring, by the device, compression history allocation, compression fragmentation and compression ratios of each of the plurality of instances of the WAN optimizer.

5. The method of claim 1, wherein step (b) further comprises monitoring, by the device, or more of the following of each of the plurality of instances of the WAN optimizer: resource utilization, number of connections, number of claims and bandwidth usage.

6. The method of claim 1, wherein step (c) further comprises determining, by the device, that a metric computed from monitoring network traffic has exceeded a threshold and responsive to the determination, automatically selecting by the device a second load balancing scheme to load balance the plurality of instances of the WAN optimizer.

7. The method of claim 1, wherein step (c) further comprises automatically switching, by the device, from the load balancing scheme to the selected load balancing scheme while executing the plurality of instances of the WAN optimizer.

8. The method of claim 1, further comprising automatically changing, by the device responsive to the monitoring, the number of instances of the WAN optimizer executing on the device.
9. The method of claim 1, further comprising automatically adjusting, by the device responsive to the monitoring, a size of resource usage used by one or more of the plurality of instances of the WAN optimizer.
10. The method of claim 1, further comprises applying, by the device one or more rules to data collected from monitoring, to determine to change one or more of the following: a number of instances of the WAN optimizer, a size of one or more WAN optimizers and the load balancing scheme.
11. A system for managing a plurality of instances of a Wide Area Network (WAN) optimizer executing on an intermediary device, the system comprising:

a device intermediary to a plurality of clients and a plurality of servers;

a plurality of instances of a Wide Area Network (WAN) optimizer executing on the device to accelerate WAN communications between the plurality of clients and the plurality of servers;

a monitor that monitors network traffic traversing the device for each of the plurality of instances of the WAN optimizer; and

a manager executing on the device that, responsive to the monitor, selects a change of a load balancing scheme to load balance the plurality of instances of the WAN optimizer.

12. The system of claim 11, wherein the manager automatically establishes a configuration of a size of each of the plurality of instances of the WAN optimizer based on data stored from monitoring of previous execution of the plurality of instances of the WAN optimizer.
13. The system of claim 11, wherein each of the plurality of instances of the WAN optimizer execute as a virtual machine in a virtualized environment.
14. The system of claim 11, wherein the monitor monitors compression history allocation, compression fragmentation and compression ratios of each of the plurality of instances of the WAN optimizer.
15. The system of claim 11, wherein the monitor monitors one or more of the following of each of the plurality of instances of the WAN optimizer: resource utilization, number of connections, number of claims and bandwidth usage.

16. The system of claim 11, wherein the manager determines that a metric computed from monitoring network traffic has exceeded a threshold and responsive to the determination, automatically selects a second load balancing scheme to load balance the plurality of instances of the WAN optimizer.

17. The system of claim 11, wherein the manager automatically switches from a current load balancing scheme to the selected load balancing scheme while executing the plurality of instances of the WAN optimizer.

18. The system of claim 11, wherein the manager, responsive to the monitor, automatically changes the number of instances of the WAN optimizer executing on the device.

19. The system of claim 11, wherein the manager automatically adjusts, responsive to the monitor, a size of resource usage used by one or more of the plurality of instances of the WAN optimizer.

20. The system of claim 11, wherein the manager applies one or more rules to data collected from monitoring, to determine to change one or more of the following: a number of instances of the WAN optimizer, a size of one or more WAN optimizers and the load balancing scheme.

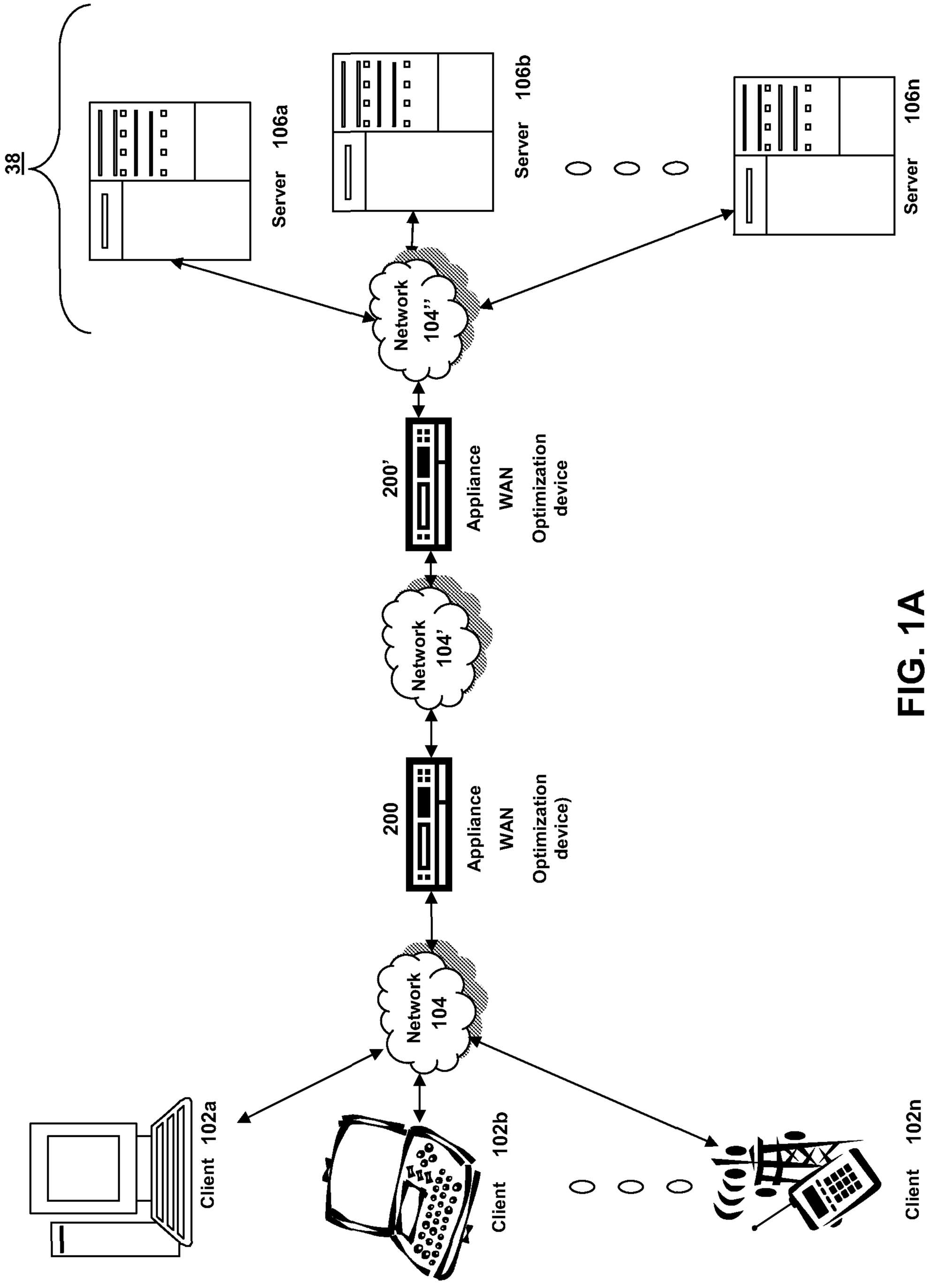
38
Server 106a
Server 106b
Server 106n
Network 104''
200'
Appliance WAN Optimization device
Network 104'
200
Appliance WAN Optimization device)
Network 104
Client 102a
Client 102b
Client 102n

FIG. 1A

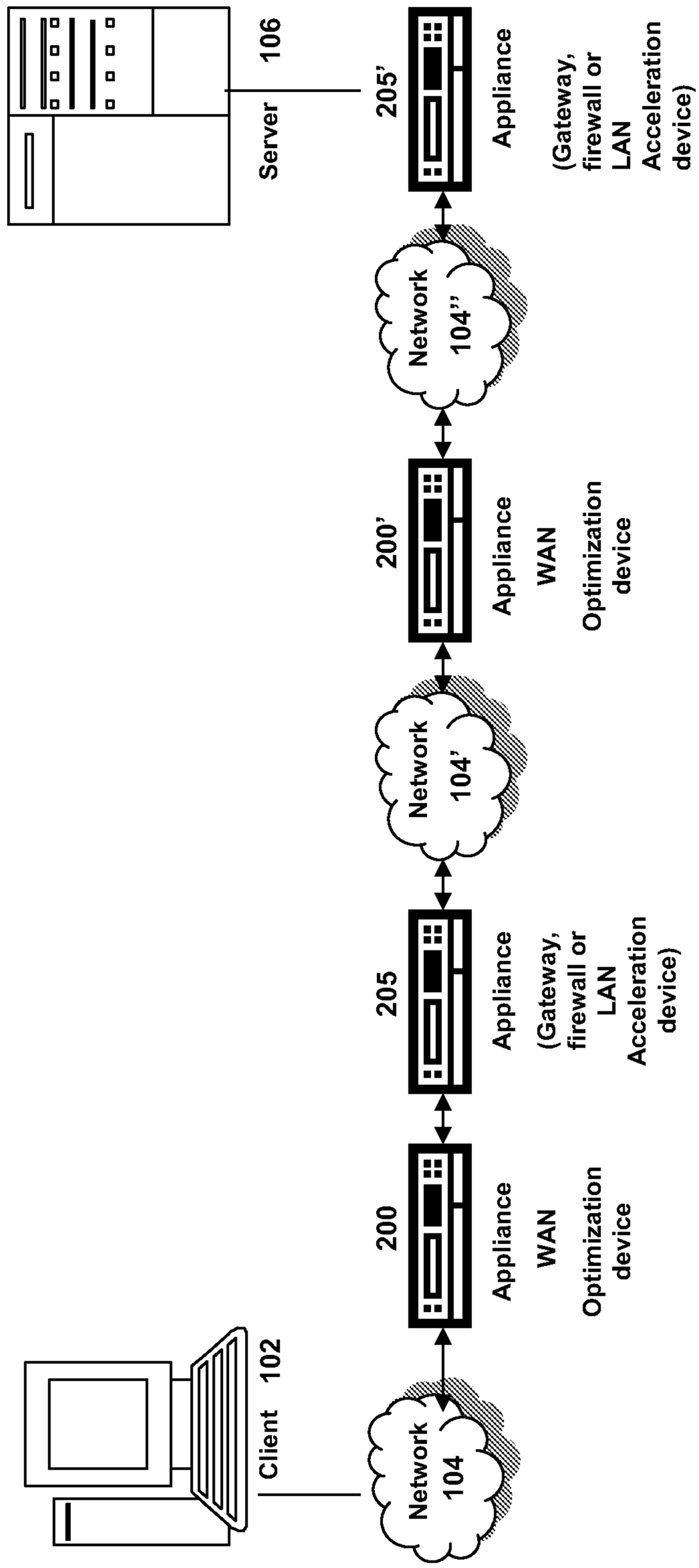
Client 102
Network 104
Appliance 200
WAN Optimization device
Network 104'
Appliance 205
(Gateway, firewall or LAN Acceleration device)
Network 104''
Appliance 200'
WAN Optimization device
Appliance 205'
(Gateway, firewall or LAN Acceleration device)
Server 106

FIG. 1B

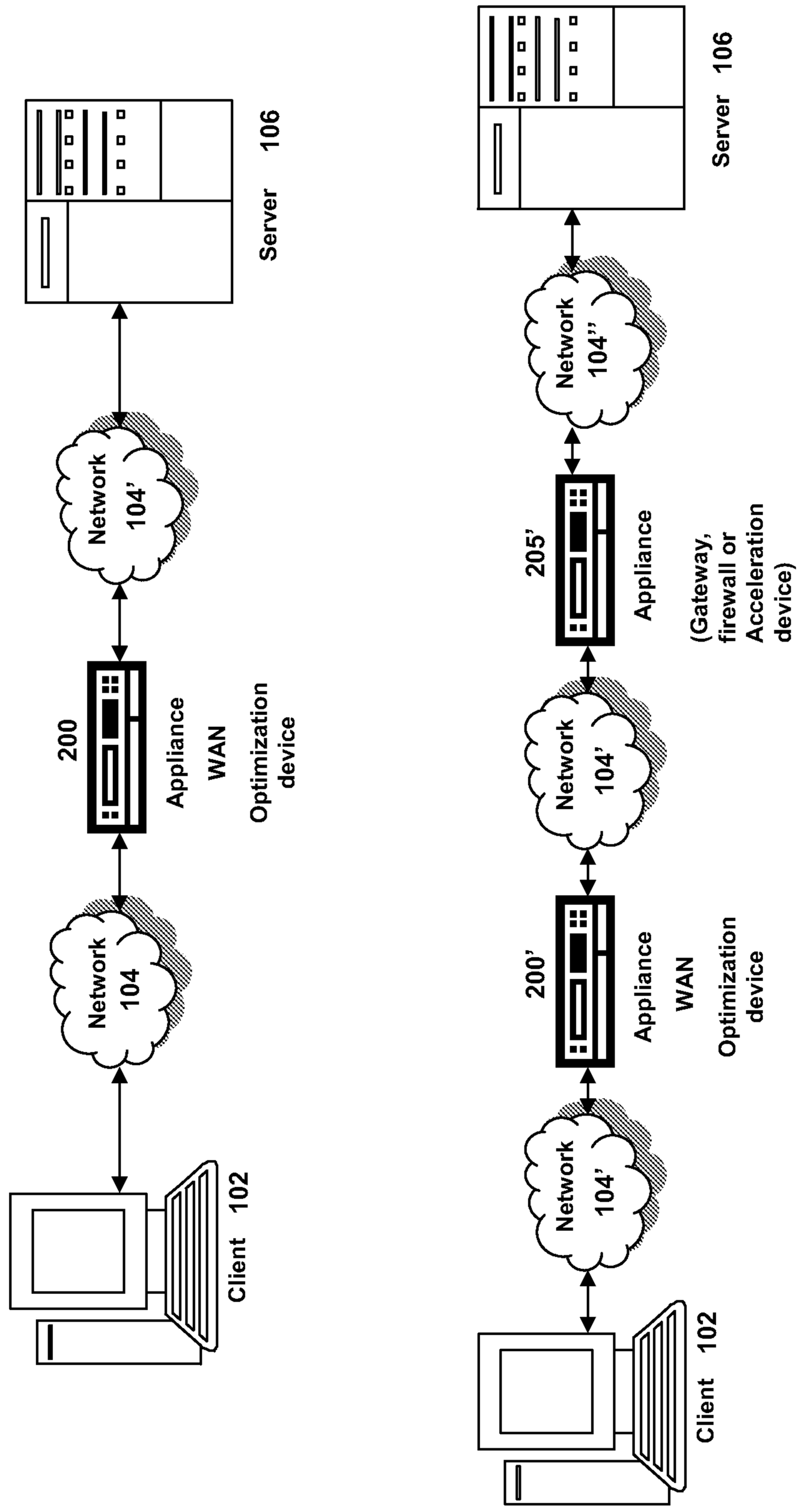
Server 106
Network 104'
205'
Appliance
(Gateway, firewall or Acceleration device)
Network 104''
Network 104'
200'
Appliance
WAN Optimization device
Network 104'
Client 102
Server 106
Network 104'
200
Appliance
WAN Optimization device
Network 104
Client 102

FIG. 1C

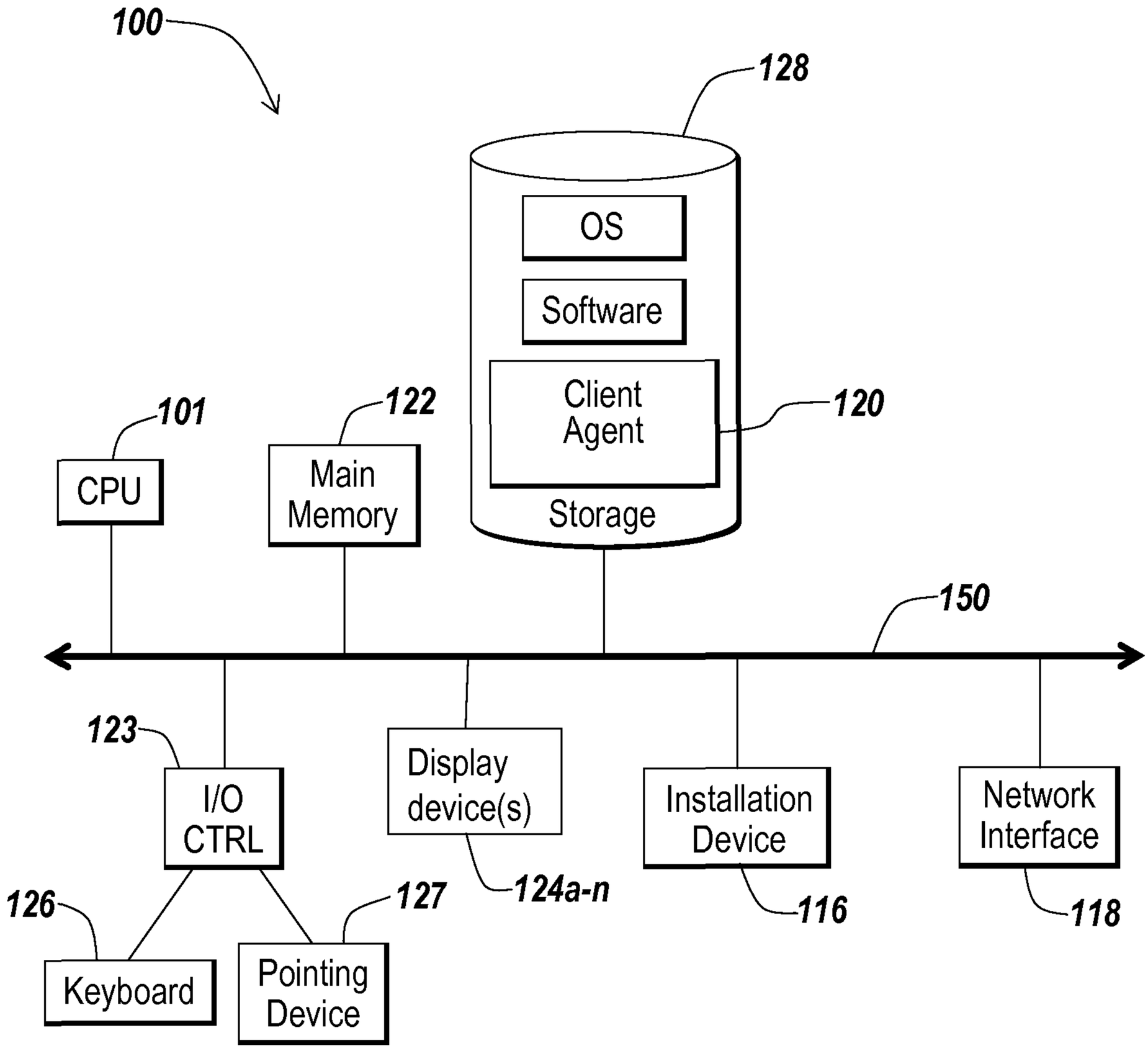
100
128
OS
Software
Client Agent
120
Storage
101
CPU
122
Main Memory
150
123
I/O CTRL
Display device(s)
124a-n
Installation Device
116
Network Interface
118
126
Keyboard
127
Pointing Device

Fig. 1D

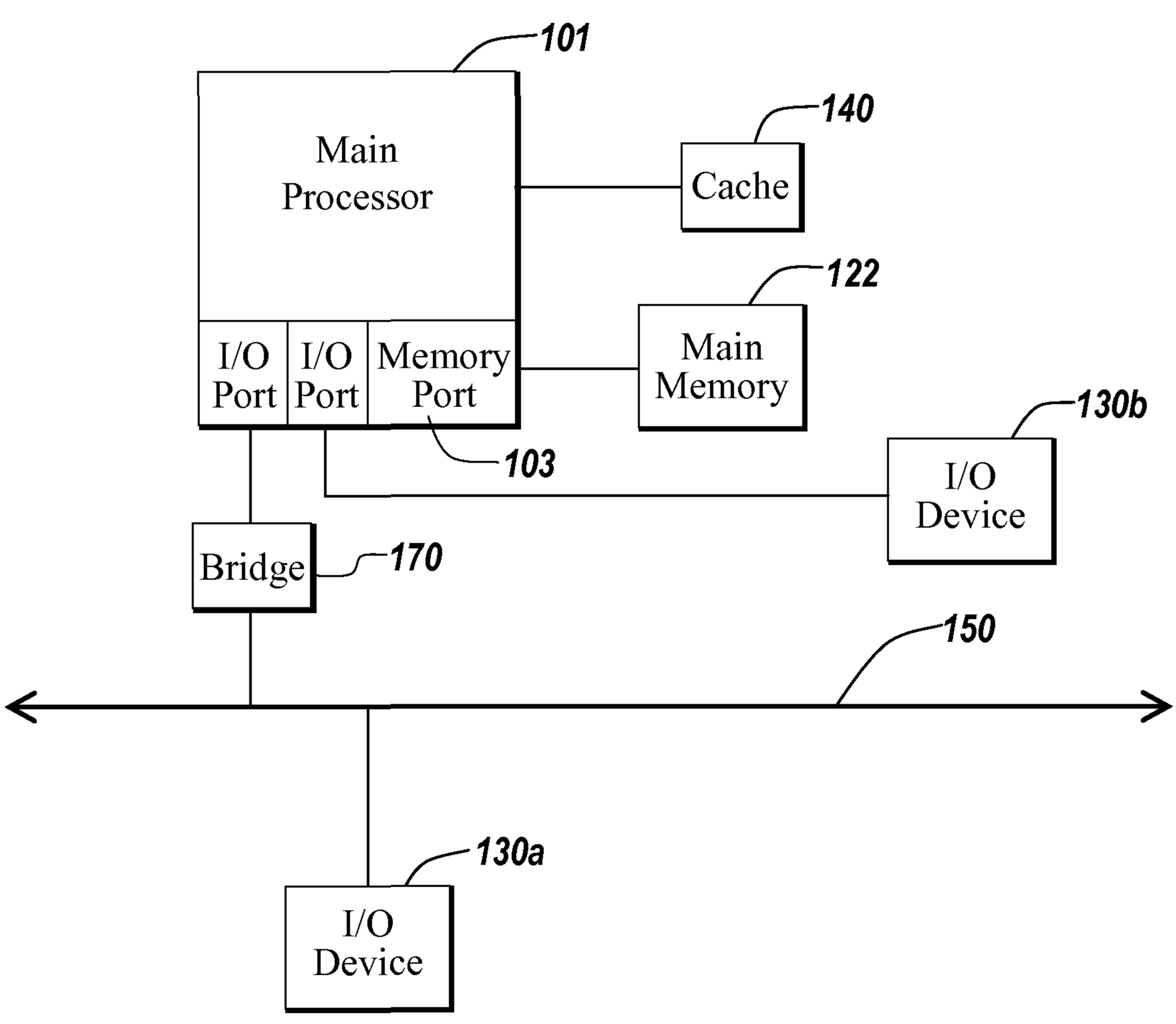
101
Main Processor
140
Cache
122
Main Memory
I/O Port
I/O Port
Memory Port
103
130b
I/O Device
Bridge
170
150
130a
I/O Device

Fig. 1E

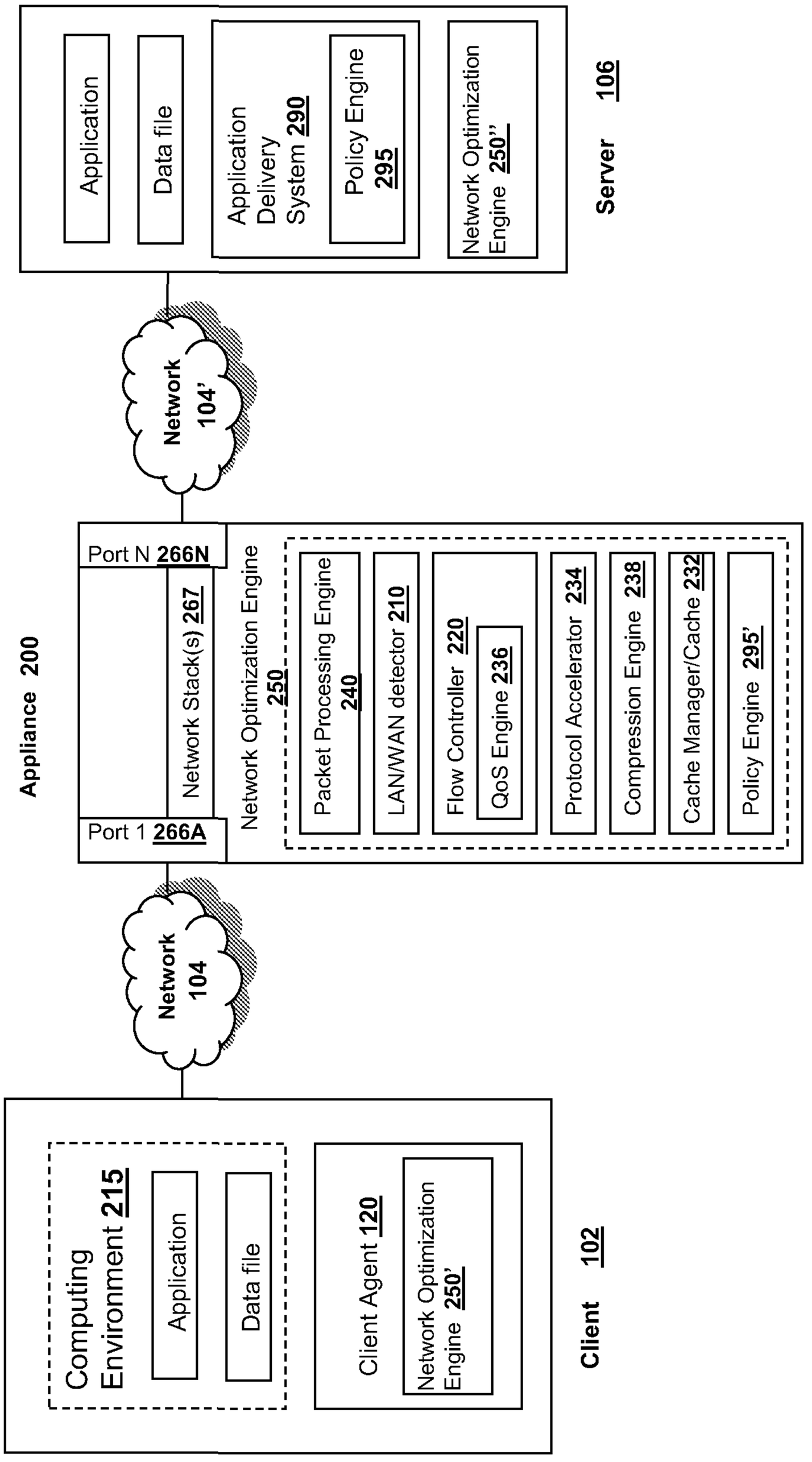
Server 106
Application
Data file
Application Delivery System 290
Policy Engine 295
Network Optimization Engine 250''
Network 104'
Appliance 200
Port N 266N
Network Stack(s) 267
Network Optimization Engine 250
Packet Processing Engine 240
LAN/WAN detector 210
Flow Controller 220
QoS Engine 236
Protocol Accelerator 234
Compression Engine 238
Cache Manager/Cache 232
Policy Engine 295'
Port 1 266A
Network 104
Computing Environment 215
Application
Data file
Client Agent 120
Network Optimization Engine 250'
Client 102

FIG. 2A

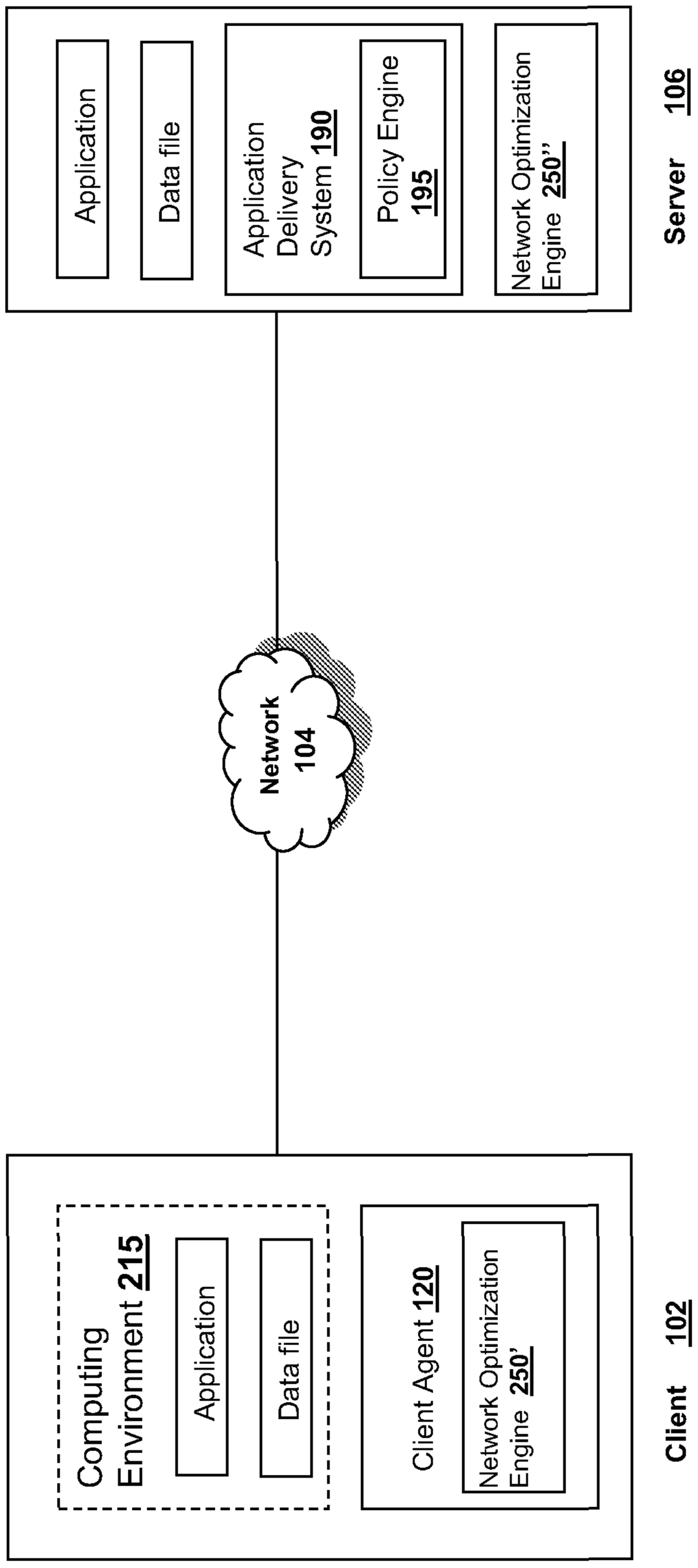
Application
Data file
Application Delivery System 190
Policy Engine 195
Network Optimization Engine 250''
Server 106
Network 104
Computing Environment 215
Application
Data file
Client Agent 120
Network Optimization Engine 250'
Client 102

FIG. 2B

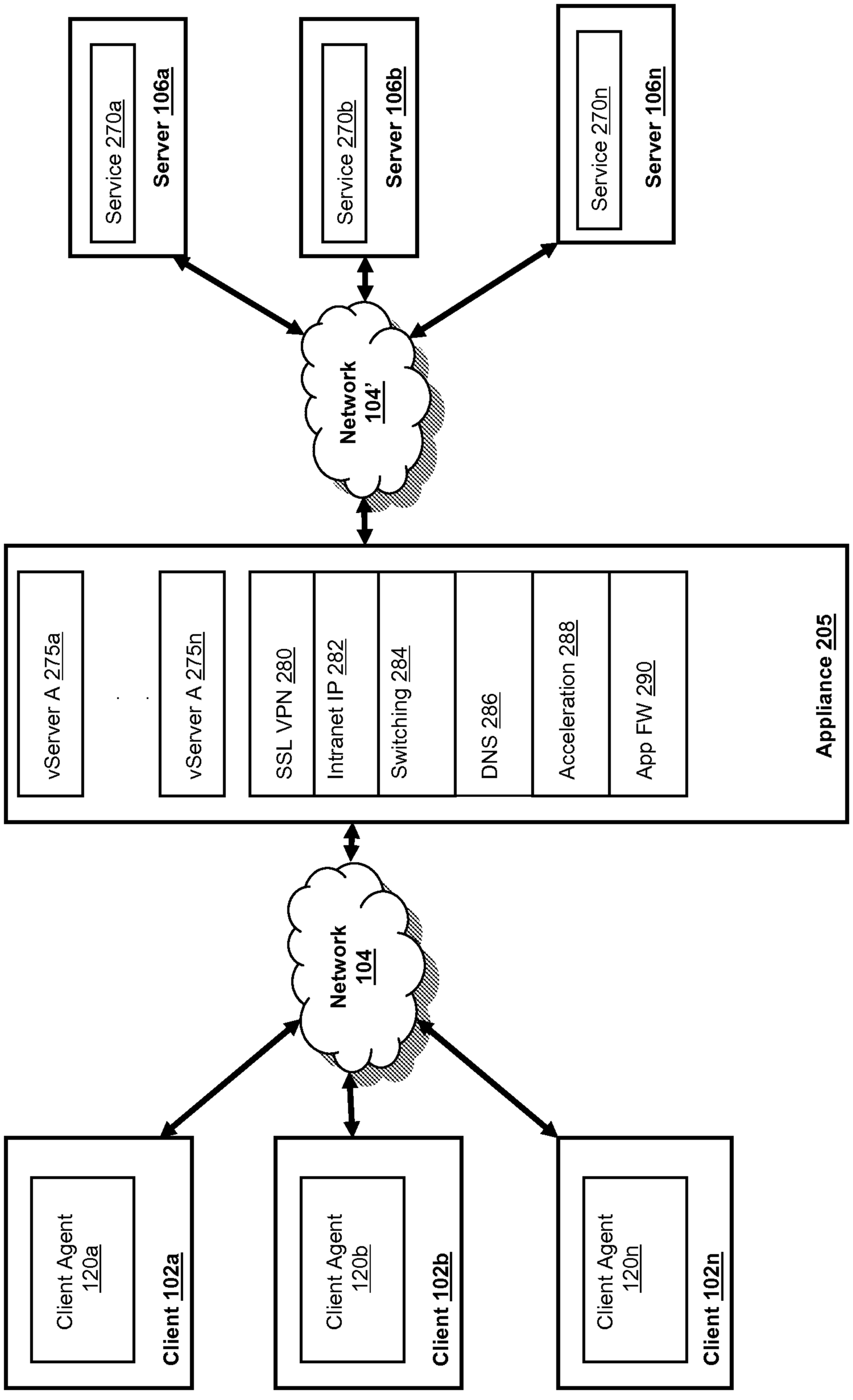
Client 102a
Client Agent 120a
Client 102b
Client Agent 120b
Client 102n
Client Agent 120n
Network 104
Appliance 205
vServer A 275a
vServer A 275n
SSL VPN 280
Intranet IP 282
Switching 284
DNS 286
Acceleration 288
App FW 290
Network 104'
Server 106a
Service 270a
Server 106b
Service 270b
Server 106n
Service 270n

FIG. 2C

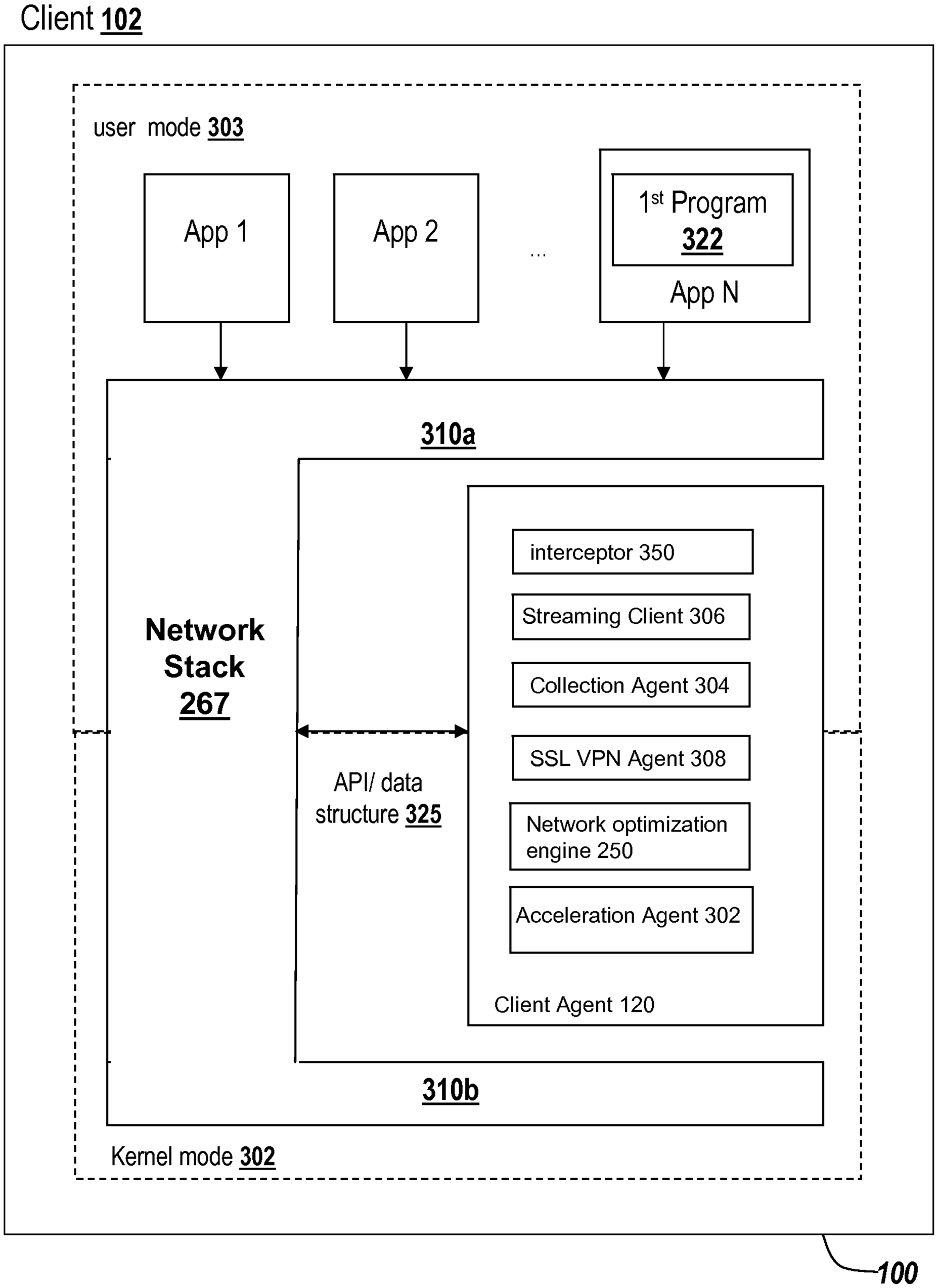
Client 102
user mode 303
App 1
App 2
...
1st Program 322
App N
310a
Network Stack 267
interceptor 350
Streaming Client 306
Collection Agent 304
SSL VPN Agent 308
API/ data structure 325
Network optimization engine 250
Acceleration Agent 302
Client Agent 120
310b
Kernel mode 302
100

Fig. 3

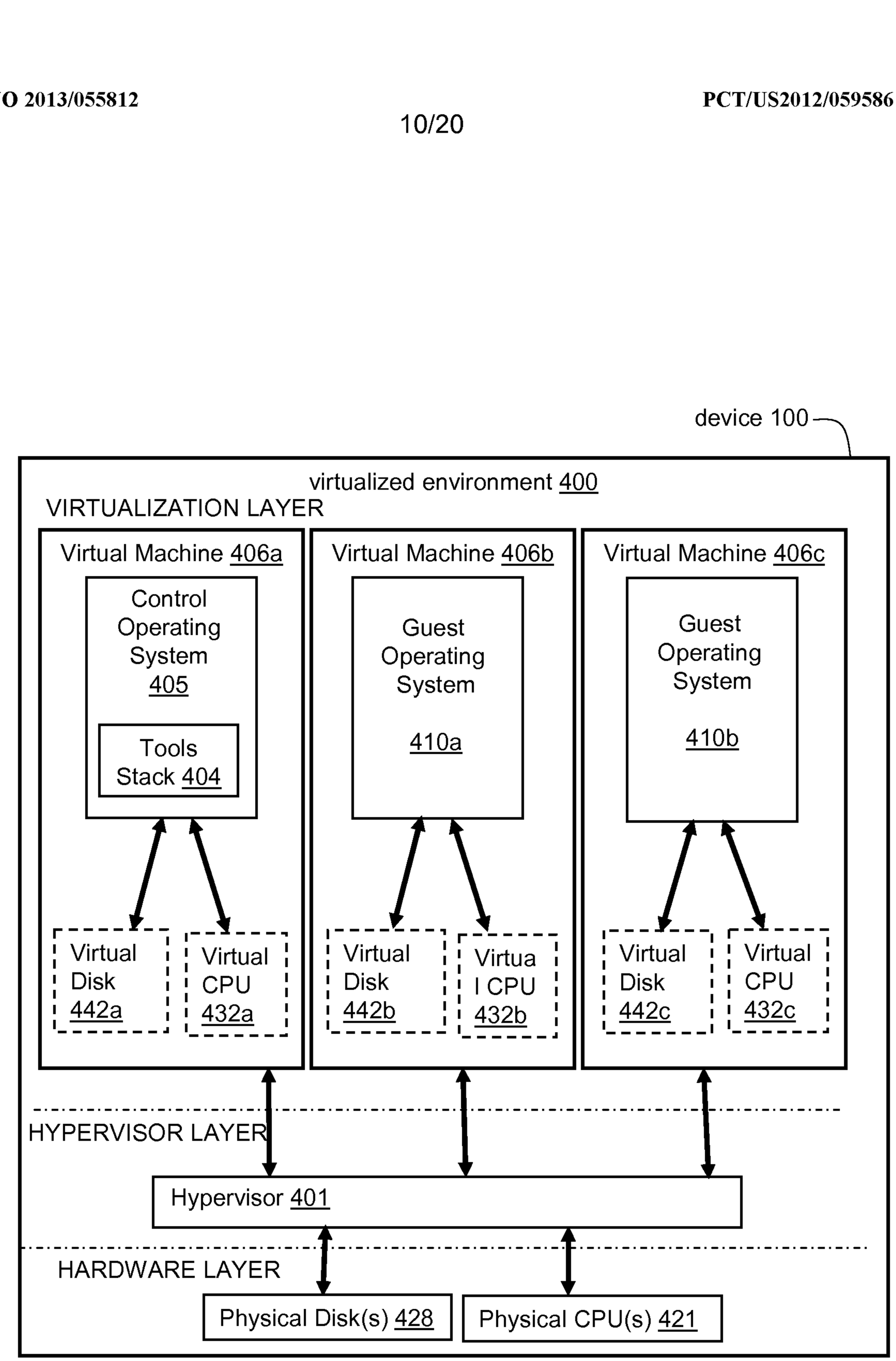
device 100
virtualized environment 400
VIRTUALIZATION LAYER
Virtual Machine 406a
Control Operating System 405
Tools Stack 404
Virtual Disk 442a
Virtual CPU 432a
Virtual Machine 406b
Guest Operating System 410a
Virtual Disk 442b
Virtua l CPU 432b
Virtual Machine 406c
Guest Operating System 410b
Virtual Disk 442c
Virtual CPU 432c
HYPERVISOR LAYER
Hypervisor 401
HARDWARE LAYER
Physical Disk(s) 428
Physical CPU(s) 421

FIG. 4A

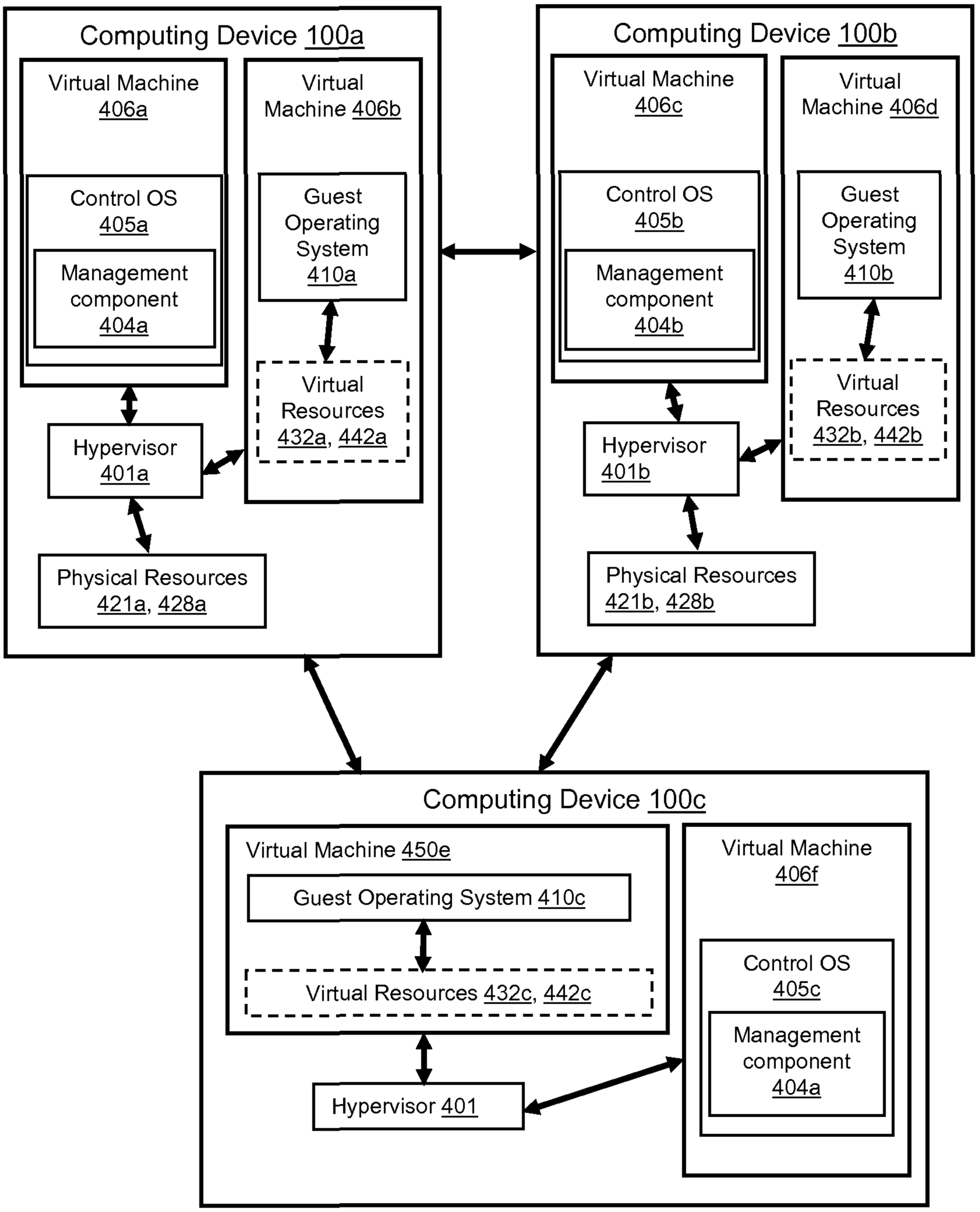
Computing Device 100a
Virtual Machine 406a
Control OS 405a
Management component 404a
Virtual Machine 406b
Guest Operating System 410a
Virtual Resources 432a, 442a
Hypervisor 401a
Physical Resources 421a, 428a
Computing Device 100b
Virtual Machine 406c
Control OS 405b
Management component 404b
Virtual Machine 406d
Guest Operating System 410b
Virtual Resources 432b, 442b
Hypervisor 401b
Physical Resources 421b, 428b
Computing Device 100c
Virtual Machine 450e
Guest Operating System 410c
Virtual Resources 432c, 442c
Hypervisor 401
Virtual Machine 406f
Control OS 405c
Management component 404a

FIG. 4B

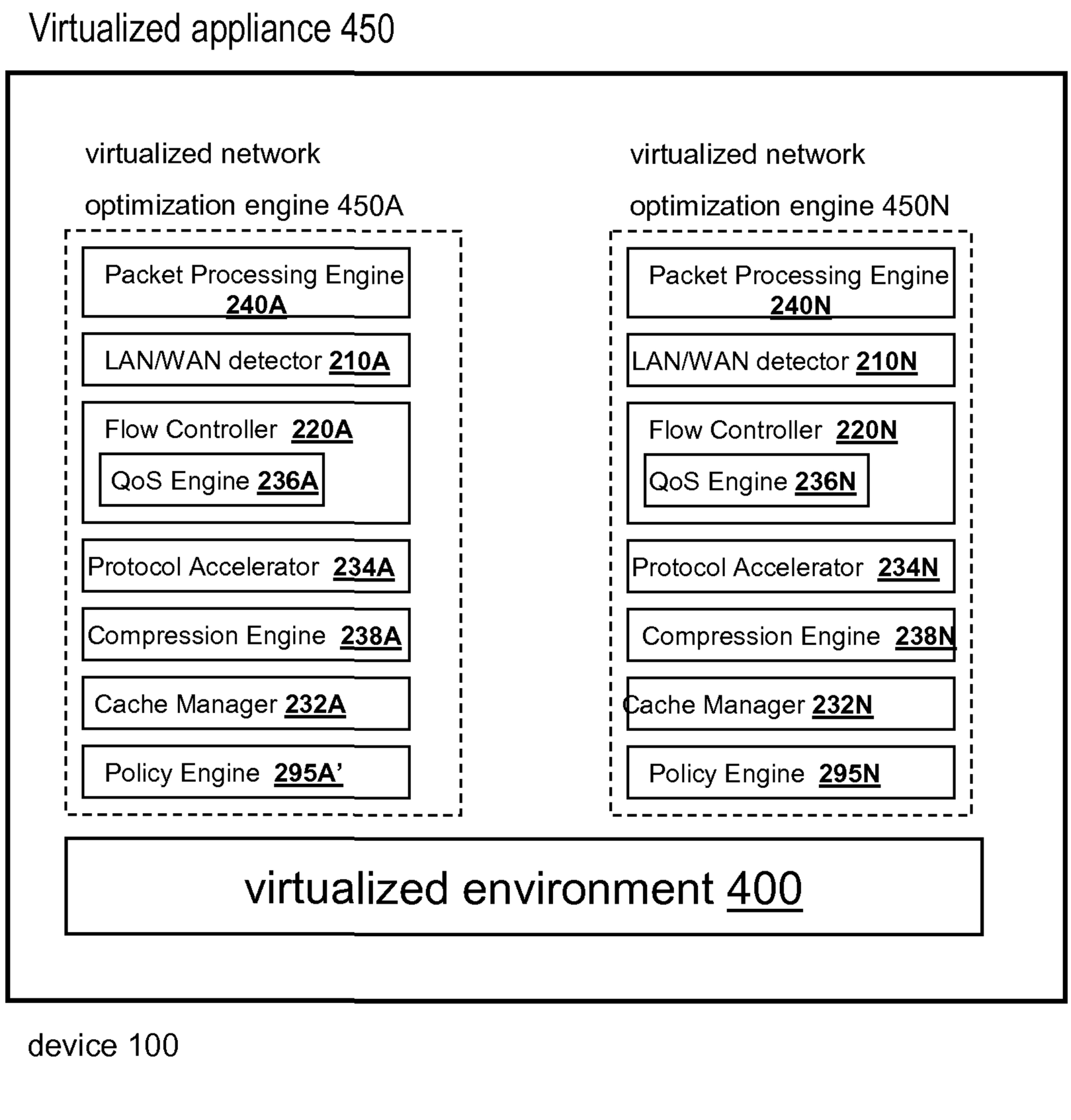
Virtualized appliance 450
virtualized network optimization engine 450A
Packet Processing Engine 240A
LAN/WAN detector 210A
Flow Controller 220A
QoS Engine 236A
Protocol Accelerator 234A
Compression Engine 238A
Cache Manager 232A
Policy Engine 295A'
virtualized network optimization engine 450N
Packet Processing Engine 240N
LAN/WAN detector 210N
Flow Controller 220N
QoS Engine 236N
Protocol Accelerator 234N
Compression Engine 238N
Cache Manager 232N
Policy Engine 295N
virtualized environment 400
device 100

FIG. 4C

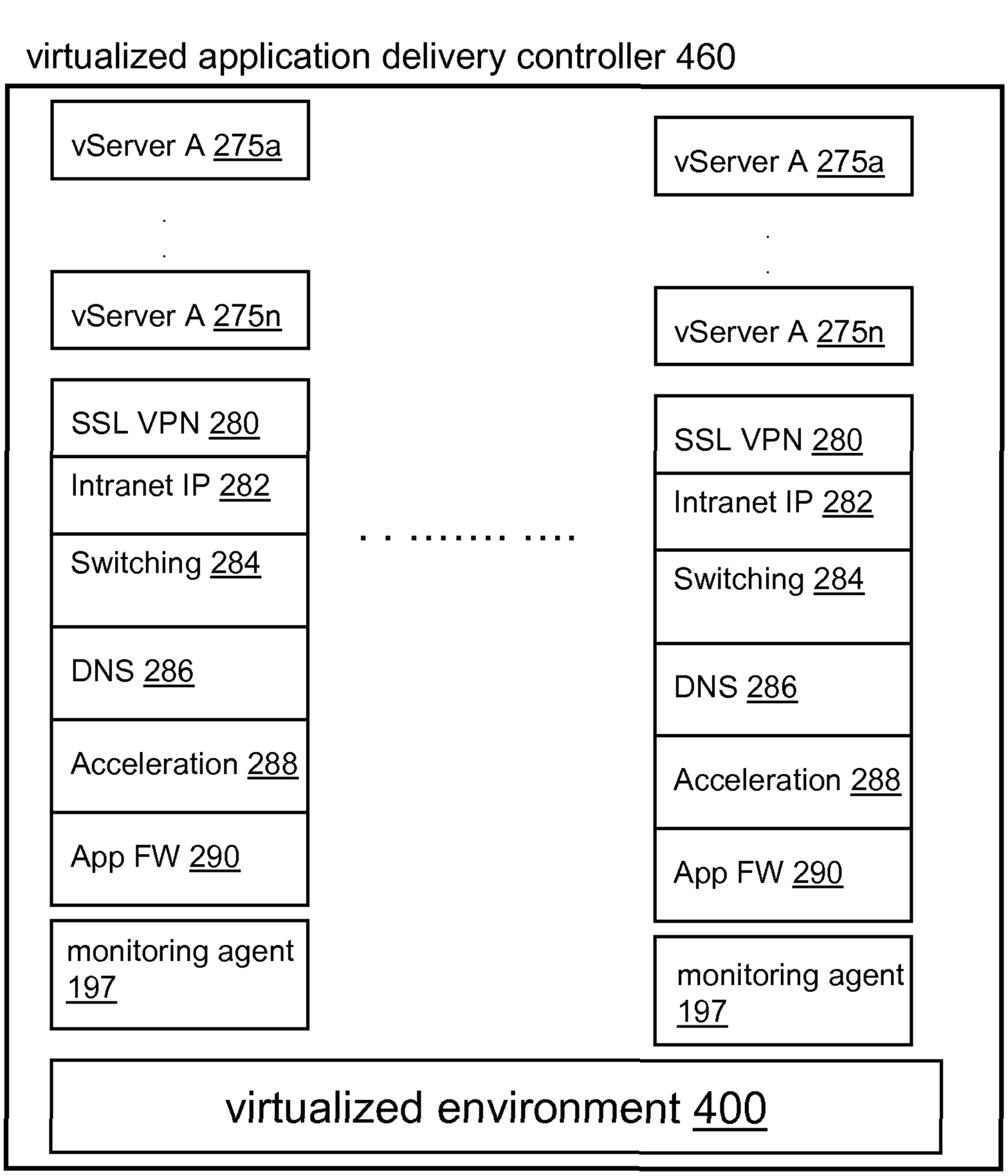
virtualized application delivery controller 460
vServer A 275a
vServer A 275n
SSL VPN 280
Intranet IP 282
Switching 284
DNS 286
Acceleration 288
App FW 290
monitoring agent 197
vServer A 275a
vServer A 275n
SSL VPN 280
Intranet IP 282
Switching 284
DNS 286
Acceleration 288
App FW 290
monitoring agent 197
virtualized environment 400
computing device 100

FIG. 4D

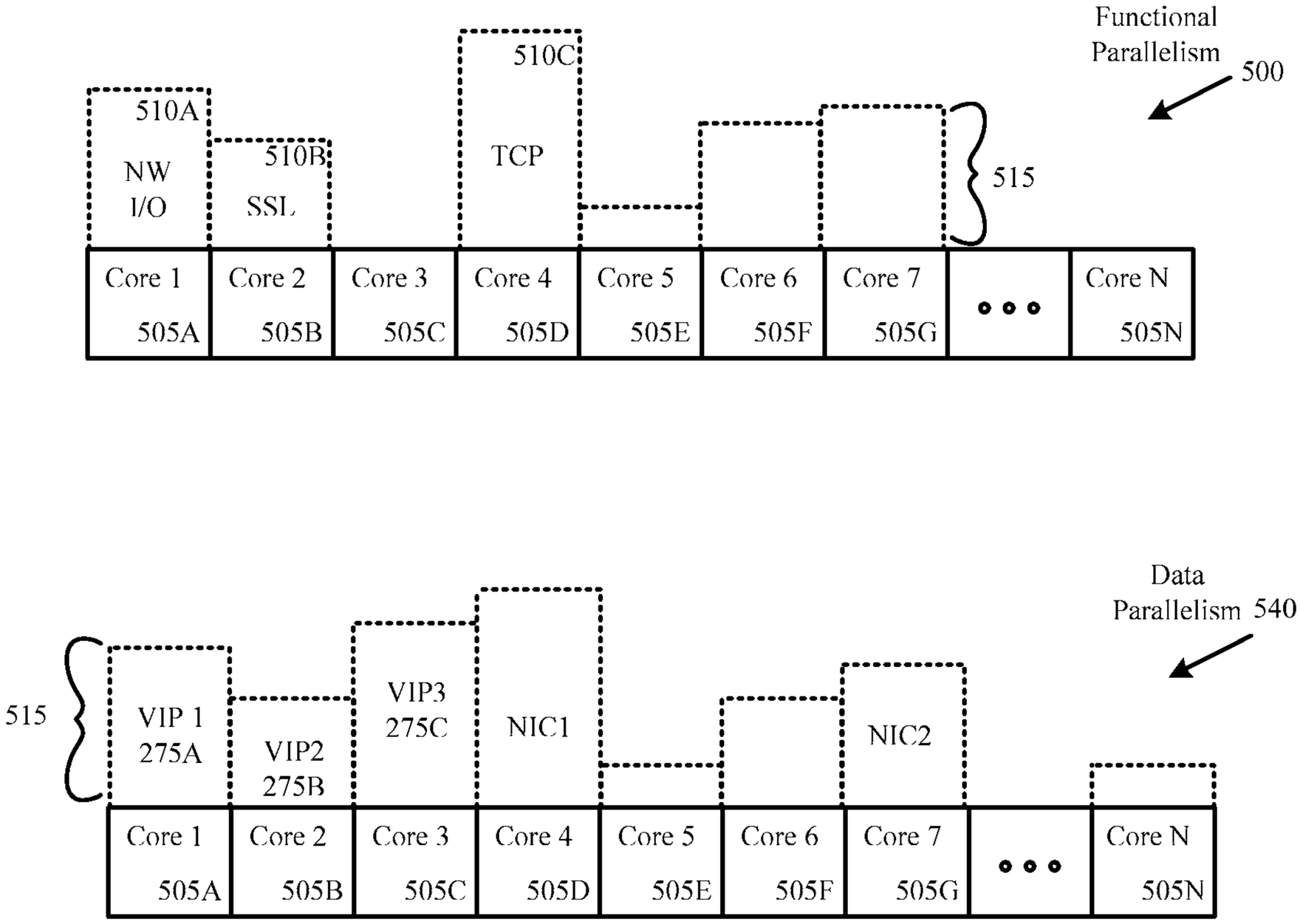
Functional Parallelism 500
510A
NW I/O
510B
SSL
510C
TCP
515
Core 1 505A
Core 2 505B
Core 3 505C
Core 4 505D
Core 5 505E
Core 6 505F
Core 7 505G
Core N 505N
Data Parallelism 540
515
VIP 1 275A
VIP2 275B
VIP3 275C
NIC1
NIC2
Core 1 505A
Core 2 505B
Core 3 505C
Core 4 505D
Core 5 505E
Core 6 505F
Core 7 505G
Core N 505N
Flow-Based Data Parallelism 520
515
515A
515B
515C
515D
515E
515F
515G
515N
Core 1 505A
Core 2 505B
Core 3 505C
Core 4 505D
Core 5 505E
Core 6 505F
Core 7 505G
Core N 505N

FIG. 5A

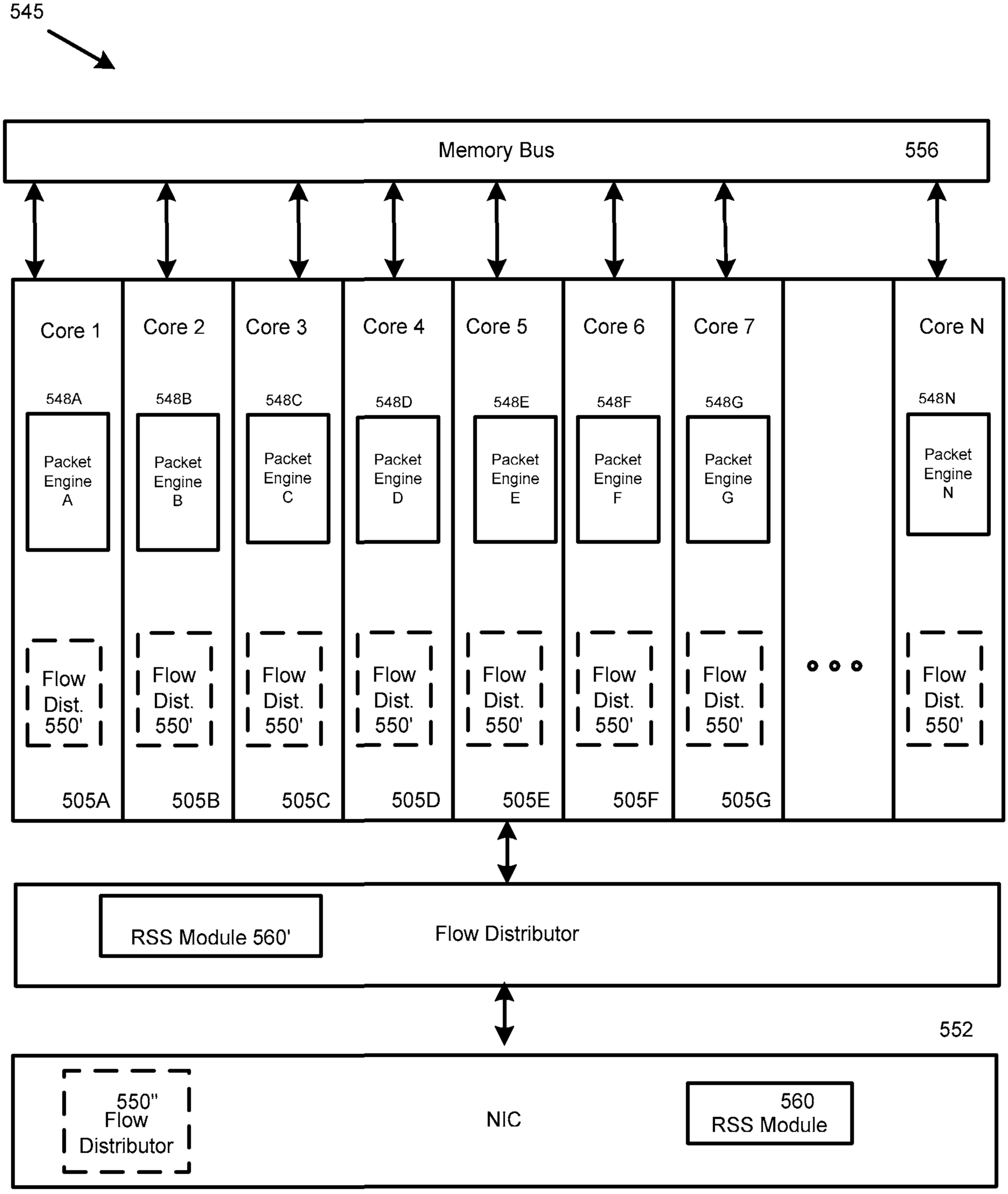
545
Memory Bus
556
Core 1
Core 2
Core 3
Core 4
Core 5
Core 6
Core 7
Core N
548A
548B
548C
548D
548E
548F
548G
548N
Packet Engine A
Packet Engine B
Packet Engine C
Packet Engine D
Packet Engine E
Packet Engine F
Packet Engine G
Packet Engine N
Flow Dist. 550'
505A
505B
505C
505D
505E
505F
505G
RSS Module 560'
Flow Distributor
552
550" Flow Distributor
NIC
560 RSS Module

FIG. 5B

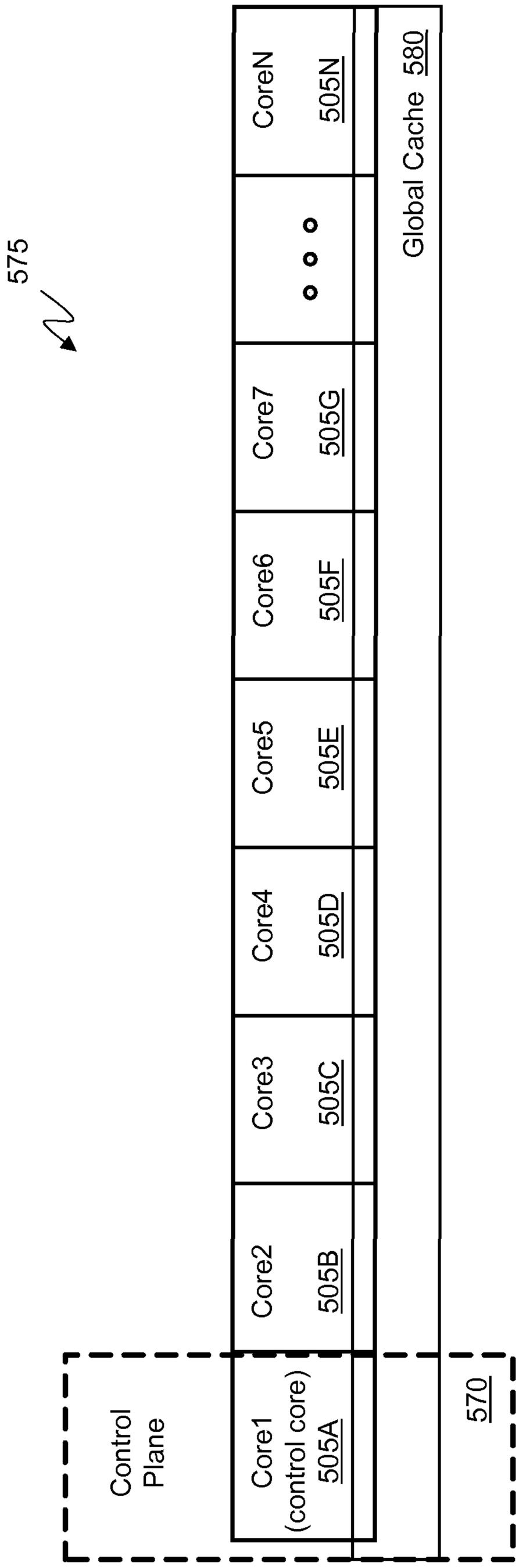
575
CoreN
505N
Core7
505G
Core6
505F
Core5
505E
Core4
505D
Core3
505C
Core2
505B
Core1
(control core)
505A
Control
Plane
Global Cache 580
570

FIG. 5C

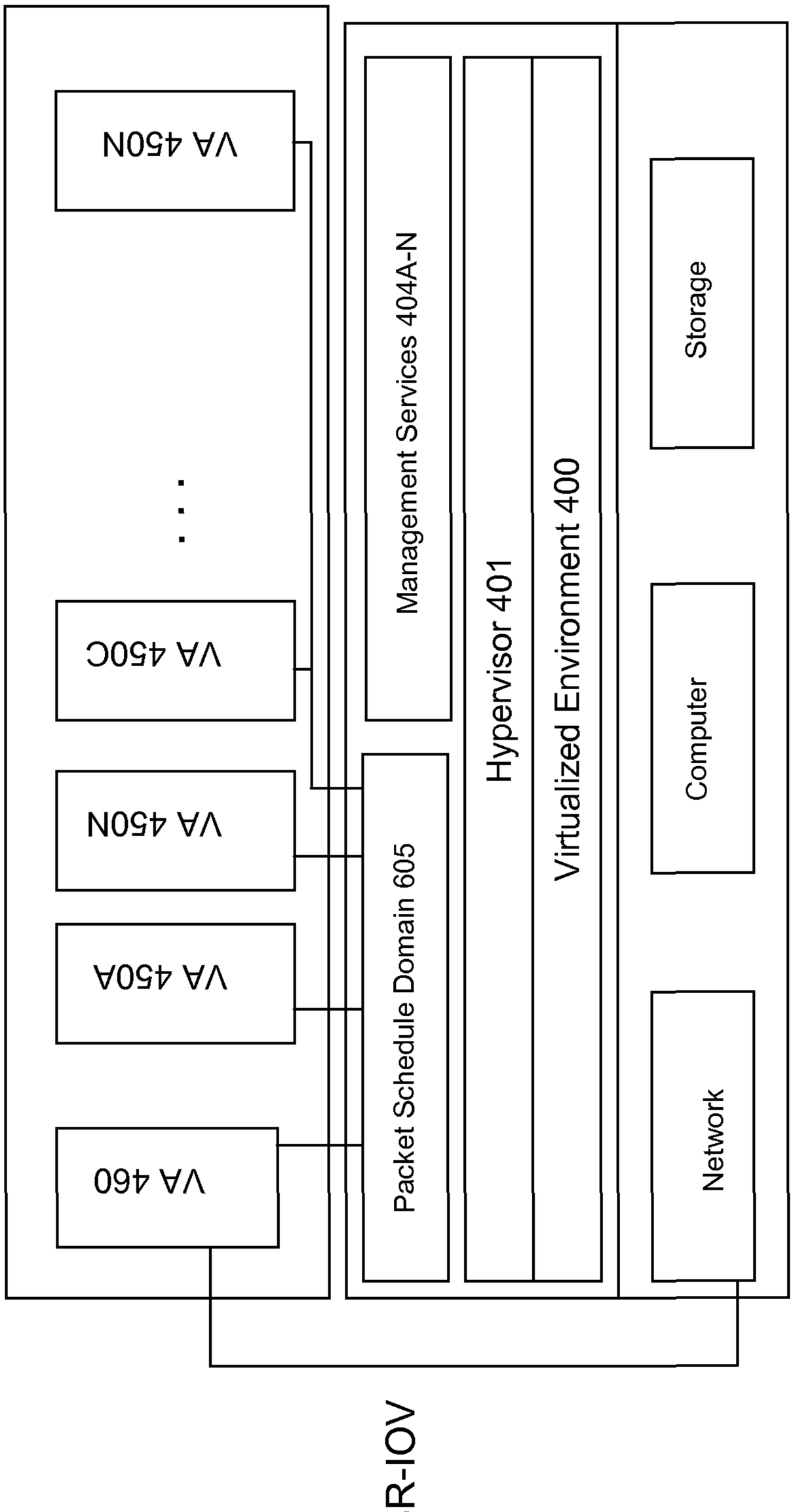
VA 450N
VA 450C
VA 450N
VA 450A
VA 460
Packet Schedule Domain 605
Management Services 404A-N
Hypervisor 401
Virtualized Environment 400
Storage
Computer
Network
Appliance 200
SR-IOV

FIG. 6A

	Load Balancing Method	Balance Effectiveness	Compression History	Bandwidth Management	Simplicity	Fault Tolerant	Persistent Unit
1	Least connection (sub-optimal, easiest to configure)	↑↑	↓↓	?	↑↑	↑↑	Client
2	Static configuration (optimal WAN optimization but difficult to manage)	↑	↑↑	→	↓↓	↓↓ Single spare option is available	Branch
3	Least accumulated load using receive bandwidth + AgentID persistence (better WAN optimization characteristics)	↑	↑↑	↑↑	→	↑↑	Branch
4	Least connection + AgentID persistence (requires branches to be same size,)	↓	↑	↑↑	↑	↑	Branch
5	Source IP hash to load up one BR at a time	?	↑	↑	↑↑	↑↑	Branch

Legend:
↑↑ Best
↑ Better
→ Neutral
↓ Poor
↓↓ Worst
? Don't know

FIG. 6B

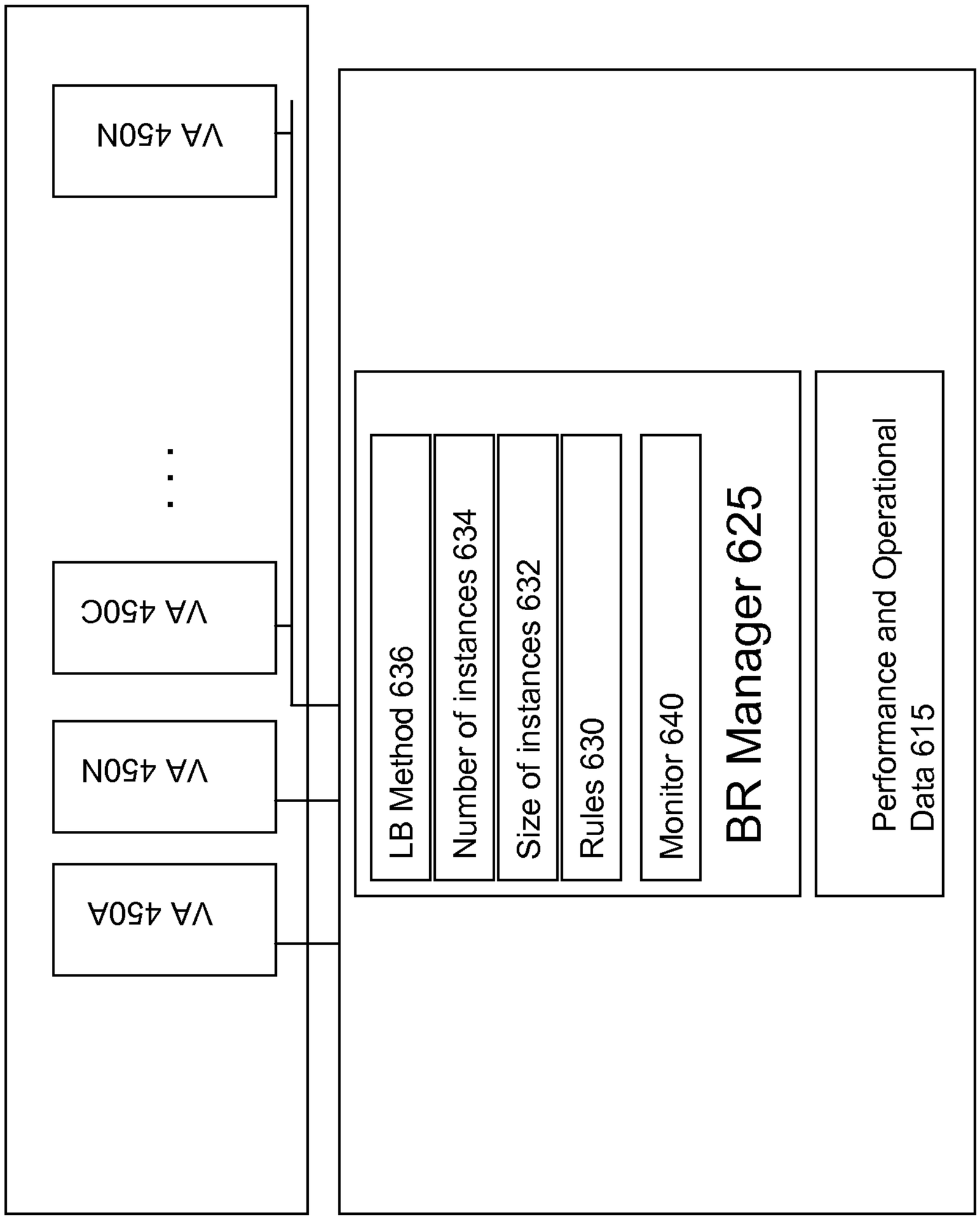
Appliance 200
VA 450A
VA 450C
VA 450N
VA 450N
Performance and Operational Data 615
BR Manager 625
LB Method 636
Number of instances 634
Size of instances 632
Rules 630
Monitor 640

FIG. 6C

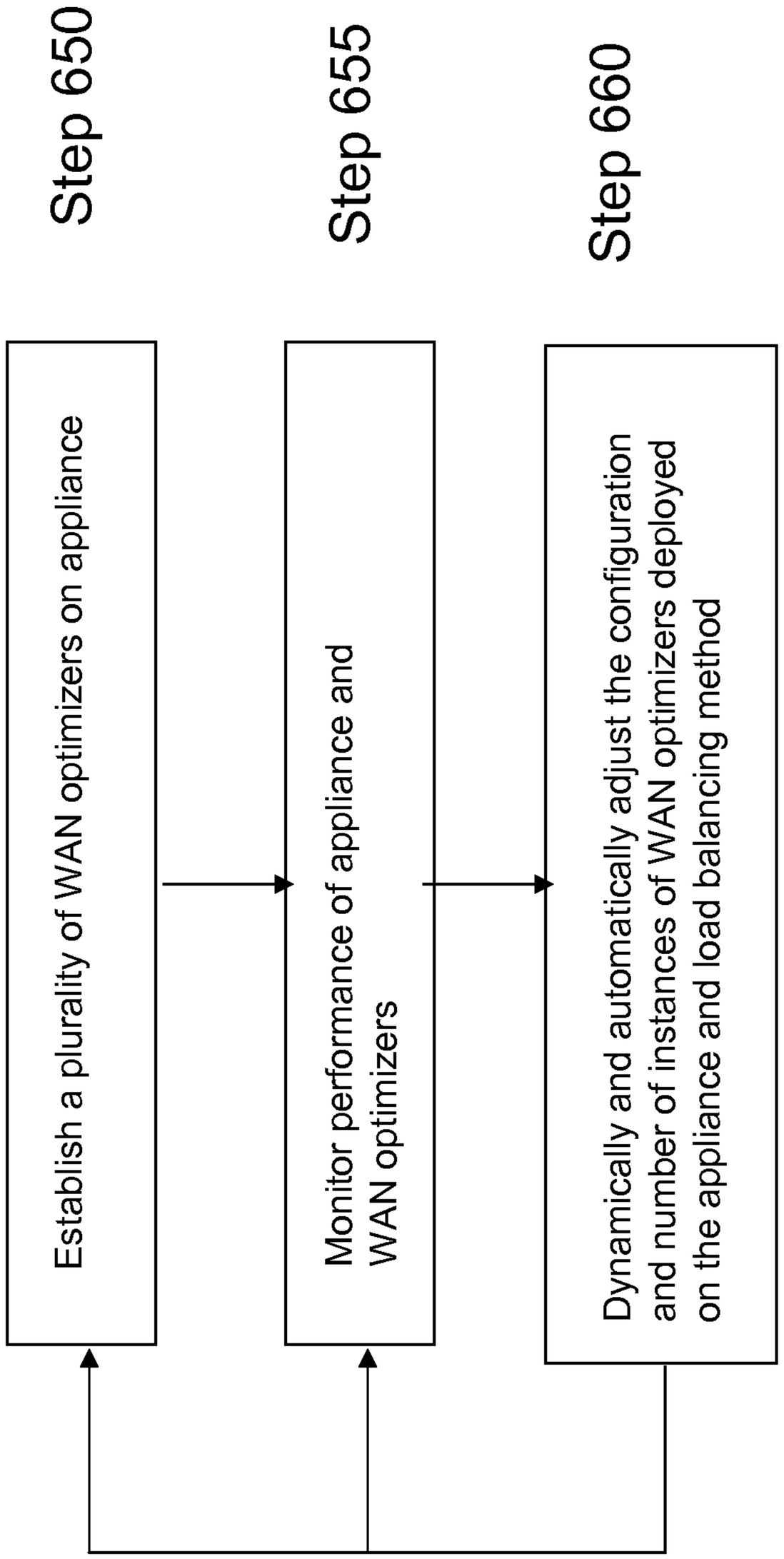
Establish a plurality of WAN optimizers on appliance
Step 650
Monitor performance of appliance and WAN optimizers
Step 655
Dynamically and automatically adjust the configuration and number of instances of WAN optimizers deployed on the appliance and load balancing method
Step 660

FIG. 6D

SYSTEMS AND METHODS FOR DYNAMIC ADAPTATION OF NETWORK ACCELERATORS

Related Application

This application claims the benefit of and priority to U.S. Provisional Application No. 61/547,493, entitled "Systems and Methods For Dynamic Adaptation of Network Accelerators" and filed on October 14, 2011, which is incorporated herein by reference for all purposes.

Field of the Invention

The present disclosure generally relates to data communication networks. In particular, the present disclosure relates to systems and methods for the dynamic adaptation of network accelerators on a platform.

Background of the Invention

Traditional network elements have been developed and deployed as discrete network appliances and functions. As such, they have been deployed in a particular way and sized to accommodate some sort of network traffic model. The model can be as simple as "I need gigabit speed for this link" to more sophisticated models where traffic analysis and peak loading are measured and placed in a model that determines network element sizing.

Brief Summary of the Invention

As computing in general and networking in particular moves to more virtualized environments, there exist several problems with the prior models. As the workload become more mobile and dynamic, traditional network engineering becomes virtually impossible and new mechanisms must be used to "adapt" the network infrastructure to the offered load.

In some deployments, a load balancer may be used to load balance multiple WAN optimizers. In further deployments, a virtualized load balancer may be used to load balance multiple virtualized WAN optimizers The load balancer may be configured to use or apply any of a plurality of load balancing methods, such as but not limited to a least connection method, least connection with agent id persistence, a static configuration, least accumulated load using receive bandwidth, least accumulated load using receive bandwidth with agent id

persistence and source internet protocol (IP) hashing. Each of these methods may provide varying load balance effectiveness, compression history performance, bandwidth management and simplicity with respect to load balancing WAN optimizers. There may be no one-size-fits-all solution for all deployments. There may be a trade off between ease of deployment and WAN optimization.

Systems and methods of the present solution provide a more optimal solution by dynamically and automatically reacting to changing network workload. A system that starts slowly, either by just examining traffic passively or by doing sub-optimal acceleration can learn over time, how many peer WAN optimizers are being serviced by an appliance, how much traffic is coming from each peer WAN optimizers, and the type of traffic being seen. Knowledge from this learning can serve to provide a better or improved baseline for the configuration of an appliance. In some embodiments, based on resources (e.g., CPU, Memory, Disk), the system from this knowledge may determine how many WAN optimization instances should be used and of what size, and how the load should be distributed across the instances of the WAN optimizer. Some example rules are as follows:

1. If a small number of peer WAN optimizers exist, a smaller number of large WAN optimizer instances should be provisioned on an appliance because compression histories will be less fragmented and compression ratios higher (better)
2. When peers WAN optimizers are of significantly different sizes, they should be distributed unevenly across the WAN optimizers instances (perhaps using WAN optimizers instances of different sizes)

Over time the system may continue to monitor the workload for changes and change the load distribution as needed, for example, adding additional WAN optimizers instances because first pass compression (high CPU utilization) is being used extensively and compression history fragmentation effects would be minimal. Or traffic from a remote site has grown significantly and a larger WAN optimizers instance is required to keep compression history from fragmenting.

In one aspects, the present solution is directed to a method for managing a plurality of instances of a Wide Area Network (WAN) optimizer executing on an intermediary device. The method includes establishing, on a device intermediary to a plurality of clients and plurality of servers, a plurality of instances of a Wide Area Network (WAN) optimizer to accelerate WAN communications between the plurality of clients and the plurality of servers.

The method includes monitoring, by the device, network traffic traversing the device for each of the plurality of instances of the WAN optimizer and selecting, by a manager executing on the device responsive to the monitoring, a change of a load balancing scheme to load balance the plurality of instances of the WAN optimizer.

The method may automatically establish, by the device, a configuration of a size of each of the plurality of instances of the WAN optimizer based on data stored from monitoring of previous execution of the plurality of instances of the WAN optimizer. Each of the plurality of instances of the WAN optimize may execute as a virtual machine in a virtualized environment.

In some embodiments, the method includes comprises monitoring, by the device, compression history allocation, compression fragmentation and compression ratios of each of the plurality of instances of the WAN optimizer. In some embodiments, the method includes monitoring, by the device, or more of the following of each of the plurality of instances of the WAN optimizer: resource utilization, number of connections, number of claims and bandwidth usage.

In some embodiments, the method includes determining, by the device, that a metric computed from monitoring network traffic has exceeded a threshold and responsive to the determination, automatically selecting by the device a second load balancing scheme to load balance the plurality of instances of the WAN optimizer. In some embodiments, the method includes automatically switching, by the device, from the load balancing scheme to the selected load balancing scheme while executing the plurality of instances of the WAN optimizer. In some embodiments, the method includes automatically changing, by the device responsive to the monitoring, the number of instances of the WAN optimizer executing on the device. In some embodiments, the method includes automatically adjusting, by the device responsive to the monitoring, a size of resource usage used by one or more of the plurality of instances of the WAN optimizer. The method may also include applying, by the device one or more rules to data collected from monitoring, to determine to change one or more of the following: a number of instances of the WAN optimizer, a size of one or more WAN optimizers and the load balancing scheme.

In some aspects, the present solution is directed to a system for managing a plurality of instances of a Wide Area Network (WAN) optimizer executing on an intermediary device. The system includes a device intermediary to a plurality of clients and plurality of servers and a plurality of instances of a Wide Area Network (WAN) optimizer executing on the device to accelerate WAN communications between the plurality of clients and the plurality of servers;