

# Technical Disclosure Commons

---

Defensive Publications Series

---

September 2023

## DATA BATCHING FOR REDUCTION OF TRANSMISSION BURSTS

Siddharth Ray  
*Meta Platforms, Inc.*

Amit Shukla  
*Meta Platforms, Inc.*

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Ray, Siddharth and Shukla, Amit, "DATA BATCHING FOR REDUCTION OF TRANSMISSION BURSTS", Technical Disclosure Commons, (September 13, 2023)  
[https://www.tdcommons.org/dpubs\\_series/6246](https://www.tdcommons.org/dpubs_series/6246)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## **DATA BATCHING FOR REDUCTION OF TRANSMISSION BURSTS**

### **TECHNICAL FIELD**

**[0001]** The present description generally relates to computer networks and, more particularly, to systems and methods of batching data for transmission over a computer network.

### **BACKGROUND**

**[0002]** Computing devices may connect to computer networks to communicate with other computing devices via one or more network interfaces, such as Wi-Fi or Ethernet. Network transmissions (e.g., via Wi-Fi) may be bursty and send data periodically as opposed to continuously.

### **BRIEF SUMMARY**

**[0003]** An aspect of the subject technology may include a method that includes receiving a first control signal, closing a gate corresponding to a buffer in response to the first control signal, receiving one or more data packets with the buffer, receiving a second control signal, opening the gate corresponding to the buffer in response to the second control signal, and transferring one or more data packets of the buffer for transmission in response to the gate opening.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0004]** Certain features of the subject technology are set forth in the appended claims. However, for purpose of explanation, several embodiments of the subject technology are set forth in the following figures.

**[0005]** FIG. 1 illustrates an example network environment, in accordance with one or more examples of the subject technology.

**[0006]** FIG. 2 illustrates an example system for carrying out the subject technology, in accordance with one or more examples of the subject technology.

**[0007]** FIG. 3 illustrates a block diagram of an example software stack of the electronic device, in accordance with one or more examples of the subject technology.

**[0008]** FIG. 4 illustrates a flow diagram of an exemplary process for data batching, in accordance with one or more examples of the subject technology.

**[0009]** FIG. 5 illustrates an artificial reality system comprising a head-mounted display in accordance with one or more examples of the subject technology.

### **DETAILED DESCRIPTION**

**[0010]** The detailed description set forth below is intended as a description of various configurations of the subject technology and is not intended to represent the only configurations in which the subject technology may be practiced. The appended drawings are incorporated herein and constitute a part of the detailed description. The detailed description includes specific details for the purpose of providing a thorough understanding of the subject technology. However, the subject technology is not limited to the specific details set forth herein and may be practiced using one or more other embodiments of the subject technology. In one or more embodiments of the subject technology, structures and components are shown in block diagram form in order to avoid obscuring the concepts of the subject technology.

**[0011]** As defined herein a “computer-readable storage medium,” which refers to a non-transitory, physical or tangible storage medium (e.g., volatile or non-volatile memory device), may be differentiated from a “computer-readable transmission medium,” which refers to an electromagnetic signal.

**[0012]** Computing devices may connect to computer networks to communicate with other computing devices via one or more network interfaces, such as Wi-Fi or Ethernet. Network transmissions (e.g., via Wi-Fi) may be bursty and send data periodically as opposed to continuously. For example, the network interface may prepare for a transmission to another computing device over a computer network by waiting a period of time to fill at least some of a buffer with data before transmitting the data all at once.

**[0013]** A network interface performing such transmission bursts (or “bursts”) may have a plurality of states. States may include an active state and an inactive (or “sleep”) state. The network interface may be in a sleep state while waiting for data to arrive and transition to an active state to transmit a burst of data. Following the burst, the network interface may return to a sleep state to repeat the inactive to active cycle.

**[0014]** Transitioning between states may take a period of time (“transition time”) to complete. Such transition times may be non-trivial and may become much larger than the time it

takes to transmit the burst (“transmit time”), especially if the buffers are too small. The increase in transition times may result in increased power consumption, battery drain, and thermals. The subject technology aims to ameliorate these issues by reducing the number of bursts that take place in the course of network communication.

**[0015]** FIG. 1 illustrates an example network environment, in accordance with one or more examples of the subject technology. Not all of the depicted components may be used in all implementations, however, and one or more implementations may include additional or different components than those shown in the figure. Variations in the arrangement and type of the components may be made without departing from the spirit or scope of the claims as set forth herein. Additional components, different components, or fewer components may be provided.

**[0016]** The network environment 100 may include an electronic device 102 and an electronic device 104. The network 106 may communicatively (directly or indirectly) couple the electronic device 102 and the electronic device 104. In one or more implementations, the network 106 may be an interconnected network of devices that may include, or may be communicatively coupled to, the Internet. For explanatory purposes, the network environment 100 is illustrated in FIG. 1 as including the electronic device 102 and the electronic device 104; however, the network environment 100 may include any number of electronic devices and/or any number of servers communicatively coupled to each other directly or via the network 106.

**[0017]** The electronic device 102 may be, for example, smart glasses, a desktop computer, a server, a portable computing device such as a laptop computer, a smartphone, a peripheral device (e.g., a digital camera, headphones), a tablet device, a wearable device (e.g., a watch, a band, and/or the like), or any other appropriate device that includes, for example, one or more wireless interfaces, such as WLAN radios, cellular radios, Bluetooth radios, Zigbee radios, near field communication (NFC) radios, and/or other wireless radios. The electronic device 102 may be, and/or may include all or part of, the electronic system (e.g., electronic device 200) discussed below with respect to FIG. 2 or the artificial reality system 500 of FIG. 5. The electronic device 102 may be configured to transmit and receive data packets between the electronic device 104.

**[0018]** The electronic device 104 may be, for example, smart glasses, a desktop computer, a server, a portable computing device such as a laptop computer, a smartphone, a peripheral device (e.g., a digital camera, headphones), a tablet device, a wearable device (e.g., a watch, a band, and/or

the like), or any other appropriate device that includes, for example, one or more wireless interfaces, such as WLAN radios, cellular radios, Bluetooth radios, Zigbee radios, near field communication (NFC) radios, and/or other wireless radios. The electronic device 104 may be, and/or may include all or part of, the electronic system (e.g., electronic device 200) discussed below with respect to FIG. 2 or the artificial reality system 500 of FIG. 5. The electronic device 104 may be configured to transmit and receive data packets between the electronic device 102.

**[0019]** FIG. 2 illustrates an electronic device 200 for carrying out the subject technology, in accordance with one or more examples of the subject technology. The electronic device 200 may be, and/or may be a part of, the electronic device 102 and/or the electronic device 104. The electronic device 200 may include various types of computer-readable media and interfaces for various other types of computer-readable media. The electronic device 200 includes a bus 210, a system memory 204, a storage device 202, an input device interface 206, an output device interface 208, a network interface 212, and a processing unit 214, or subsets and variations thereof. Not all depicted components may be used in all embodiments, however, and one or more embodiments may include additional or different components than those shown in the figure. Variations in the arrangement and type of the components may be made without departing from the spirit or scope of the claims as set forth herein. Additional components, different components, or fewer components may be provided.

**[0020]** The bus 210 collectively represents all system, peripheral, and chipset buses that communicatively connect the numerous internal devices of the electronic device 200. In one or more embodiments, the bus 210 communicatively connects the processing unit 214 with the other components of the electronic device 200. From various memory units, the processing unit 214 retrieves instructions to execute and data to process in order to execute the operations of the subject disclosure. The processing unit 214 may be a controller and/or a single- or multi-core processor or processors in various embodiments.

**[0021]** The storage device 202 may be a read-and-write memory device. The storage device 202 may be a non-volatile memory unit that stores instructions and data (e.g., static and dynamic instructions and data) even when the electronic device 200 is off. In one or more embodiments, a mass-storage device (such as a magnetic or optical disk and its corresponding disk drive) may be used as the storage device 202. In one or more embodiments, a removable storage

device (such as a floppy disk, flash drive, and its corresponding disk drive) may be used as the storage device 202.

**[0022]** Like the storage device 202, the system memory 204 may be a read-and-write memory device. However, unlike the storage device 202, the system memory 204 may be a volatile read-and-write memory, such as random-access memory. The system memory 204 may store any of the instructions and data that one or more processing unit 214 may need at runtime to perform operations. In one or more embodiments, the processes of the subject disclosure are stored in the system memory 204 and/or the storage device 202. From these various memory units, the one or more processing unit 214 retrieves instructions to execute and data to process in order to execute the processes of one or more embodiments, discussed below. The system memory 204 may include one or more queues for storing data packets.

**[0023]** Embodiments within the scope of the present disclosure may be partially or entirely realized using a tangible computer-readable storage medium (or multiple tangible computer-readable storage media of one or more types) encoding one or more instructions. The tangible computer-readable storage medium also may be non-transitory in nature.

**[0024]** The computer-readable storage medium may be any storage medium that may be read, written, or otherwise accessed by a general purpose or special purpose computing device, including any processing electronics and/or processing circuitry capable of executing instructions. For example, without limitation, the computer-readable medium may include any volatile semiconductor memory (e.g., the system memory 204), such as RAM, DRAM, SRAM, T-RAM, Z-RAM, and TTRAM. The computer-readable medium also may include any non-volatile semiconductor memory (e.g., the storage device 202), such as ROM, PROM, EPROM, EEPROM, NVRAM, flash, nvSRAM, FeRAM, FeTRAM, MRAM, PRAM, CBRAM, SONOS, RRAM, NRAM, racetrack memory, FJG, and Millipede memory.

**[0025]** Further, the computer-readable storage medium may include any non-semiconductor memory, such as optical disk storage, magnetic disk storage, magnetic tape, other magnetic storage devices, or any other medium capable of storing one or more instructions. In one or more embodiments, the tangible computer-readable storage medium may be directly coupled to a computing device, while in other embodiments, the tangible computer-readable storage medium

may be indirectly coupled to a computing device, e.g., via one or more wired connections, one or more wireless connections, or any combination thereof.

**[0026]** Instructions may be directly executable or may be used to develop executable instructions. For example, instructions may be realized as executable or non-executable machine code or as instructions in a high-level language that may be compiled to produce executable or non-executable machine code. Further, instructions also may be realized as or may include data. Computer-executable instructions also may be organized in any format, including routines, subroutines, programs, data structures, objects, modules, applications, applets, functions, etc. As recognized by those of skill in the art, details including, but not limited to, the number, structure, sequence, and organization of instructions may vary significantly without varying the underlying logic, function, processing, and output.

**[0027]** While the above discussion primarily refers to microprocessors or multi-core processors that execute software, one or more embodiments are performed by one or more integrated circuits, such as ASICs or FPGAs. In one or more embodiments, such integrated circuits execute instructions that are stored on the circuit itself.

**[0028]** The bus 210 also connects to the input device interface 206 and output device interface 208. The input device interface 206 enables the system to receive inputs. For example, the input device interface 206 allows a user to communicate information and select commands on the electronic device 200. The input device interface 206 may be used with input devices such as keyboards, mice, dials, switches, sliders, and other interfaces (physical or virtual) for a user to supply information to the electronic device 200. The output device interface 208 may be used with output devices such as displays, speakers, and other interfaces (physical or virtual) for the electronic device 200 to provide information. One or more embodiments may include devices that function as both input and output devices, such as a touchscreen.

**[0029]** The bus 210 also couples the electronic device 200 to one or more networks and/or to one or more network nodes through the network interface 214. The network interface 214 may include one or more interfaces that allow the electronic device 200 to be a part of a network of computers (e.g., a local area network (LAN), a wide area network (WAN), or a network of networks (e.g., the Internet)). For example, the network interface 214 may include a network interface card (NIC) configured to exchange data with another device.

**[0030]** FIG. 3 illustrates a block diagram of an example software stack 300 of the electronic device 102, in accordance with one or more examples of the subject technology. Not all depicted components may be used in all embodiments, however, and one or more embodiments may include additional or different components than those shown in the figure. Variations in the arrangement and type of the components may be made without departing from the spirit or scope of the claims as set forth herein. Additional components, different components, or fewer components may be provided.

**[0031]** The electronic device 102 may include a software stack 300 that includes one or more software layers to allow applications and services to communicate with hardware of the electronic device 102. The software stack 300 may include a user space 304 and a kernel space 306.

**[0032]** The user space 304 may include applications/services 302 with which a user, program, operating system, and the like may interact. Applications may include software such as word processors, media players, accounting software, and the like, that may be interacted with by a user. Services may include software such as background tasks for responding to hardware events, daemons that run continuously as a background process then wake to handle periodic requests, and the like, that may be utilized by applications and/or operating systems. The applications/services 302 may also include software and/or hardware corresponding to layer seven of the Open Systems Interconnection (OSI) model, which is an application layer that includes protocols (e.g., HTTP, FTP) for applications to access a network.

**[0033]** The kernel space 306 may include one or more software modules of an operating system that manages the operations between the user space and the hardware of the electronic device 102. Software modules may include objects, functions, methods, binaries, and/or any other form of computer-readable instructions. The kernel space 306 may include a network stack 308, driver 310, and firmware 312.

**[0034]** The network stack 308 may include software and/or hardware corresponding to one or more layers of the Open Systems Interconnection (OSI) model, such as layers four through six. Layer four is a transport layer that provides protocols (e.g., TCP, UDP) for utilizing layers one to three to transmit data. Layer five is a session layer that provides protocols for maintaining network



connections and is responsible for controlling ports and sessions. Layer six is a presentation layer that ensures that data is in a usable format specified by the application layer.

**[0035]** The firmware 312 may include software that runs on a hardware device independent of an operating system. The hardware device may be a component of the network interface 212. For example, a NIC may be installed on the electronic device 102. The firmware 312 on the NIC may provide basic instructions for how the NIC should operate including how to convert bits into electrical signals for transmitting data and how to convert electrical signals to bits for receiving data. The NIC operates on layer one and two of the Open Systems Interconnection (OSI) model. Layer one is the physical layer and is responsible for transmitting and receiving unstructured data between devices, such as NICs, Ethernet hubs, and the like. Layer two is the data link layer that includes signal synchronization, error correction, flow control, and the like, and provides a data transmission link across one or several physical connections.

**[0036]** The driver 310 may include software that runs on the operating system and bridges the network stack 308 and the firmware 312. The driver 310 may include software that may be used by an operating system of the electronic device 102 to communicate with a hardware device (e.g., the NIC) via the firmware 312 and control the hardware device's behavior. The driver 310 may include a buffer 314 for queuing network traffic (e.g., messages), which may originate from the user space 304. In one or more implementations, the applications/services 302 may queue network traffic in addition to or instead of the buffer 314.

**[0037]** The driver 310 may pass network traffic from the user space 304 to the firmware 312 as network traffic is generated. The firmware 312 may activate (active state) the NIC to receive the network traffic after receipt by the driver 310 and transmit the received network traffic. After the network traffic has been transmitted, the firmware 312 may de-activate (inactive state) the NIC to wait for more network traffic to transmit. The transition that occurs in a cycle between an active state and inactive state consumes time that, when multiple cycles occur, may become longer than the transmission burst duration. This may result in increased power consumption and thermals.

**[0038]** In one or more embodiments of the subject technology, however, the driver 310 may accumulate an amount of network traffic in the buffer 314. A gate 318 may be provided to indicate to the driver 310 when to flush the batch of network traffic from the buffer 314 to the firmware 312. The transmission thread 320 ("TX thread") may be a software component (e.g.,

machine-readable instructions) that facilitate the flush (e.g., translation and/or transfer) of data between the driver 310 and the firmware 312.

**[0039]** When the gate is closed, network traffic may accumulate in the buffer 314. Conversely, when the gate is opened, the network traffic may be sent to the firmware 312 for transmission by the NIC. Lowering the frequency that the gate 318 is opened may increase the amount of network traffic that gets queued in the buffer 314. When more network traffic arrives at the firmware 312 at once, the firmware 312 may direct the NIC to transmit for a longer period of time than when each piece of network traffic arrives at the firmware 312 as they are generated. Because the NIC is active for a longer period of time, to transmit the batch of network traffic, the number of cycles (between active and inactive) may be reduced, thereby reducing power consumption and thermals.

**[0040]** The frequency that the gate 318 is opened and closed may be determined by a transmission control signal 316. The transmission control signal 316 may be any signal that directs the gate 318 to open and/or close. The frequency that the gate 318 is opened and closed may be based on applications/services 302. For example, an application, such as a timer, may open and close the gate 318 every 100ms. As another example, network traffic from an application may indicate that the network traffic is latency sensitive, such as network traffic from a video call, in which case the frequency may be increased to reduce the batching of network traffic in the buffer 314. The frequency that the gate 318 is opened and closed may also or instead be based on the host hardware. For example, the gate 318 may be opened in response to a transmission control signal 316 from the driver 310 indicating that the buffer 314 is full. If the buffer 314 is not full after a timeout period, the gate 318 may be opened to prevent unnecessary transmission delays. The foregoing factors for determining the frequency of opening and closing a buffer are merely exemplary, and other hardware and/or software factors are contemplated.

**[0041]** In one or more embodiments, the gating logic may be in the firmware. For example, driver 310 may pass network traffic to the firmware 312 as the network traffic arrives at the driver 310 from the applications/services 302. The network traffic may be queued in the firmware 312 and a gate may open the buffer based on a transmission control signal. The transmission control signal may also activate the NIC causing the network traffic in the buffer to be transmitted.

**[0042]** FIG. 4 illustrates a flow diagram of an exemplary process for data batching, in accordance with one or more examples of the subject technology. For explanatory purposes, the process 400 is primarily described herein with reference to FIGs. 1–3. However, the process 400 is not limited to the items shown in FIGs. 1–3, and one or more blocks (or operations) of the process 400 may be performed by one or more other components of other suitable devices (e.g., artificial reality system 500 of FIG. 5). Further, for explanatory purposes, the blocks of the process 400 are described herein as occurring in serial or linearly. However, multiple blocks of the process 400 may occur in parallel. In addition, the blocks of the process 400 need not be performed in the order shown and/or one or more blocks of the process 400 need not be performed and/or may be replaced by other operations. Although, the process 400 is described as being performed by the driver 310, the process 400 may be performed by any component of the electronic device 102 capable of storing, transferring, and/or transmitting network traffic.

**[0043]** In the process 400, at block 402, the driver 310 on the electronic device 102 may receive a transmission control signal 316. The transmission control signal 316 may be generated from any aspect of the electronic device 102 such as an application/service 302 and the operating system. The transmission control signal 316 may be received by any aspect of the electronic device that controls a buffer 314 containing network traffic (e.g., data packets), such as the driver 310 or the firmware 312.

**[0044]** At block 404, the driver 310 may close a gate 318 of the buffer 314. The transmission control signal 316 may direct the driver 310 to batch network traffic in the buffer 314 by closing the gate 318.

**[0045]** At block 406, the buffer 314 of the driver 310 receive one or more data packets with the buffer 314 to batch the network traffic. Batching the network traffic may queuing, stacking, or any other structure for grouping network traffic in memory.

**[0046]** At block 408, the driver 310 on the electronic device 102 may receive another transmission control signal 316. The transmission control signal 316 may be received from the same or different source as block 404. The transmission control signal 316 may be received at a predetermined period after block 404. In some embodiments, the transmission control signal 316 may be received in response to the buffer 314 being filled (e.g., saturated with network traffic).

**[0047]** At block 410, the driver 310 may open a gate 318 of the buffer 314. The transmission control signal 316 may direct the driver 310 to transfer network traffic in the buffer 314 by opening the gate 318. The frequency of opening and closing the gate 318 affects the size of the batch of network traffic being released from the buffer 314. If the gate 318 is not closed and remains open, the network traffic may not batch in the buffer 314 and may pass directly to the firmware 312 from the user space 304 via the driver 310. This may cause the firmware 312 to cycle between active and inactive states more frequently. Frequent cycles between active and inactive states may increase power consumption and thermals of the electronic device 102.

**[0048]** If the gate 318 is closed periodically, the network traffic may batch in the buffer 314, and the size of the batch is directly related to the duration that the gate 318 is closed. This may cause the firmware 312 to cycle between active and inactive states less frequently and thereby reduce power consumption and thermals of the electronic device.

**[0049]** At block 412, the driver 310 may transfer the batch of network traffic stored in the buffer 314 in response to the gate opening. The batch of network traffic may be transferred to the firmware 312, and the firmware 312 may transmit the batch of network traffic.

**[0050]** FIG. 5 illustrates an example artificial reality system 500. The artificial reality system 500 may include a head-mounted display (HMD) 510 (e.g., smart glasses) comprising a frame 512, one or more displays 514, and a computing device 508 (also referred to herein as computer 508). The displays 514 may be transparent or translucent allowing a user wearing the HMD 510 to look through the displays 514 to see the real world (e.g., real world environment) and displaying visual artificial reality content to the user at the same time. The HMD 510 may include an audio device 506 (e.g., speakers/microphones) that may provide audio artificial reality content to users. The HMD 510 may include one or more cameras 516, 518 which may capture images and/or videos of environments. In one exemplary embodiment, the HMD 510 may include a camera(s) 518 which may be a rear-facing camera tracking movement and/or gaze of a user's eyes.

**[0051]** One of the cameras 516 may be a forward-facing camera capturing images and/or videos of the environment that a user wearing the HMD 510 may view. The HMD 510 may include a microphone of the audio device 506 to capture voice input from the user. The artificial reality system 500 may further include a controller 504 comprising a trackpad and one or more buttons. The controller 504 may receive inputs from users and relay the inputs to the computing device

508. The controller may also provide haptic feedback to one or more users. The computing device 508 may be connected to the HMD 510 and the controller through cables or wireless connections. The computing device 508 may control the HMD 510 and the controller to provide the augmented reality content to and receive inputs from one or more users. In some example embodiments, the controller 504 may be a standalone controller or integrated within the HMD 510. The computing device 508 may be a standalone host computer device, an on-board computer device integrated with the HMD 510, a mobile device, or any other hardware platform capable of providing artificial reality content to and receiving inputs from users. In some exemplary embodiments, HMD 510 may include an artificial reality system/virtual reality system.

**[0052]** Those of skill in the art would appreciate that the various illustrative blocks, modules, elements, components, methods, and algorithms described herein may be implemented as electronic hardware, computer software, or combinations of both. To illustrate this interchangeability of hardware and software, various illustrative blocks, modules, elements, components, methods, and algorithms have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application. Various components and blocks may be arranged differently (e.g., arranged in a different order or partitioned differently) without departing from the scope of the subject technology.

**[0053]** It is understood that any specific order or hierarchy of blocks in the processes disclosed is an illustration of example approaches. Based upon design preferences, it is understood that the specific order or hierarchy of blocks in the processes may be rearranged or that all illustrated blocks be performed. Any of the blocks may be performed simultaneously. In one or more embodiments, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems may generally be integrated together in a single software product or packaged into multiple software products.

**[0054]** As used in this specification and any claims of this application, the terms “base station,” “receiver,” “computer,” “server,” “processor,” and “memory” all refer to electronic or

other technological devices. These terms exclude people or groups of people. For the purposes of the specification, the terms “display” or “displaying” means displaying on an electronic device.

**[0055]** As used herein, the phrase “at least one of” preceding a series of items, with the term “and” or “or” to separate any of the items, modifies the list as a whole rather than each member of the list (i.e., each item). The phrase “at least one of” does not require the selection of at least one of each item listed; rather, the phrase allows a meaning that includes at least one of any one of the items, and/or at least one of any combination of the items, and/or at least one of each of the items. By way of example, the phrases “at least one of A, B, and C” or “at least one of A, B, or C” each refers to only A, only B, or only C; any combination of A, B, and C; and/or at least one of each of A, B, and C.

**[0056]** The predicate words “configured to,” “operable to,” and “programmed to” do not imply any particular tangible or intangible modification of a subject but, rather, are intended to be used interchangeably. In one or more embodiments, a processor configured to monitor and control an operation or a component may also mean the processor being programmed to monitor and control the operation or the processor being operable to monitor and control the operation. Likewise, a processor configured to execute code may be construed as a processor programmed to execute code or operable to execute code.

**[0057]** Phrases such as an aspect, the aspect, another aspect, some aspects, one or more aspects, an implementation, the implementation, another implementation, some embodiments, one or more embodiments, an embodiment, the embodiment, another embodiment, some embodiments, one or more embodiments, a configuration, the configuration, another configuration, some configurations, one or more configurations, the subject technology, the disclosure, the present disclosure, other variations thereof and alike are for convenience and do not imply that a disclosure relating to such phrase(s) is essential to the subject technology or that such disclosure applies to all configurations of the subject technology. A disclosure relating to such phrase(s) may apply to all configurations or one or more configurations. A disclosure relating to such phrase(s) may provide one or more examples. A phrase such as an aspect or some aspects may refer to one or more aspects and vice versa, which applies similarly to other foregoing phrases.

**[0058]** The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any embodiment described herein as “exemplary” or as an “example” is not

necessarily to be construed as preferred or advantageous over other embodiments. Furthermore, to the extent that the term “include,” “have,” or the like is used in the description or the claims, such term is intended to be inclusive in a manner similar to the phrase “comprise” as “comprise” is interpreted when employed as a transitional word in a claim.

**[0059]** All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedicated to the public, regardless of whether such disclosure is explicitly recited in the claims. No claim element is to be construed under the provisions of 35 U.S.C. § 112(f), unless the element is expressly recited using the phrase “means for” or, in the case of a method claim, the element is recited using the phrase “step for.”

**[0060]** The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but are to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean “one and only one” unless specifically so stated, but rather “one or more.” Unless specifically stated otherwise, the term “some” refers to one or more. Pronouns in the masculine (e.g., his) include the feminine (e.g., her) and vice versa. Headings and subheadings, if any, are used for convenience only and do not limit the subject disclosure.

## CLAIMS

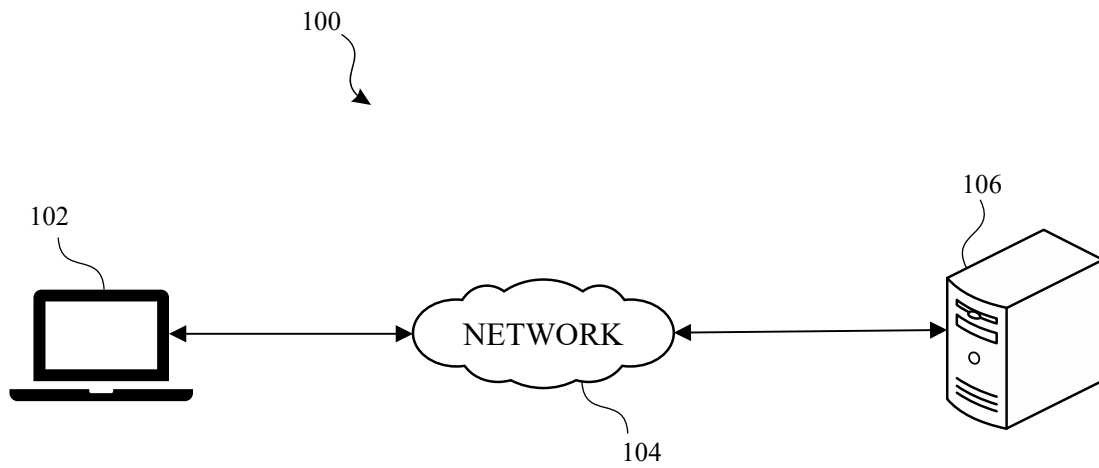
What is claimed is:

1. A method comprising:  
receiving a first control signal;  
closing a gate corresponding to a buffer in response to the first control signal;  
receiving one or more data packets with the buffer;  
receiving a second control signal;  
opening the gate corresponding to the buffer in response to the second control signal; and  
transferring one or more data packets of the buffer for transmission in response to the gate opening.
2. The method of claim 1 wherein the first control signal causes the gate to close.
3. The method of claim 1 wherein the second control signal causes the gate to open.
4. The method of claim 1 wherein the second control signal is received based on one or more of a pre-determined period of time, a buffer size, a latency associated with the one or more data packets.
5. The method of claim 1 wherein transferring one or more data packets of the buffer comprises transferring the one or more data packets to a network interface card.
6. The method of claim 1 wherein transferring one or more data packets of the buffer comprises transmitting data packets in the buffer.

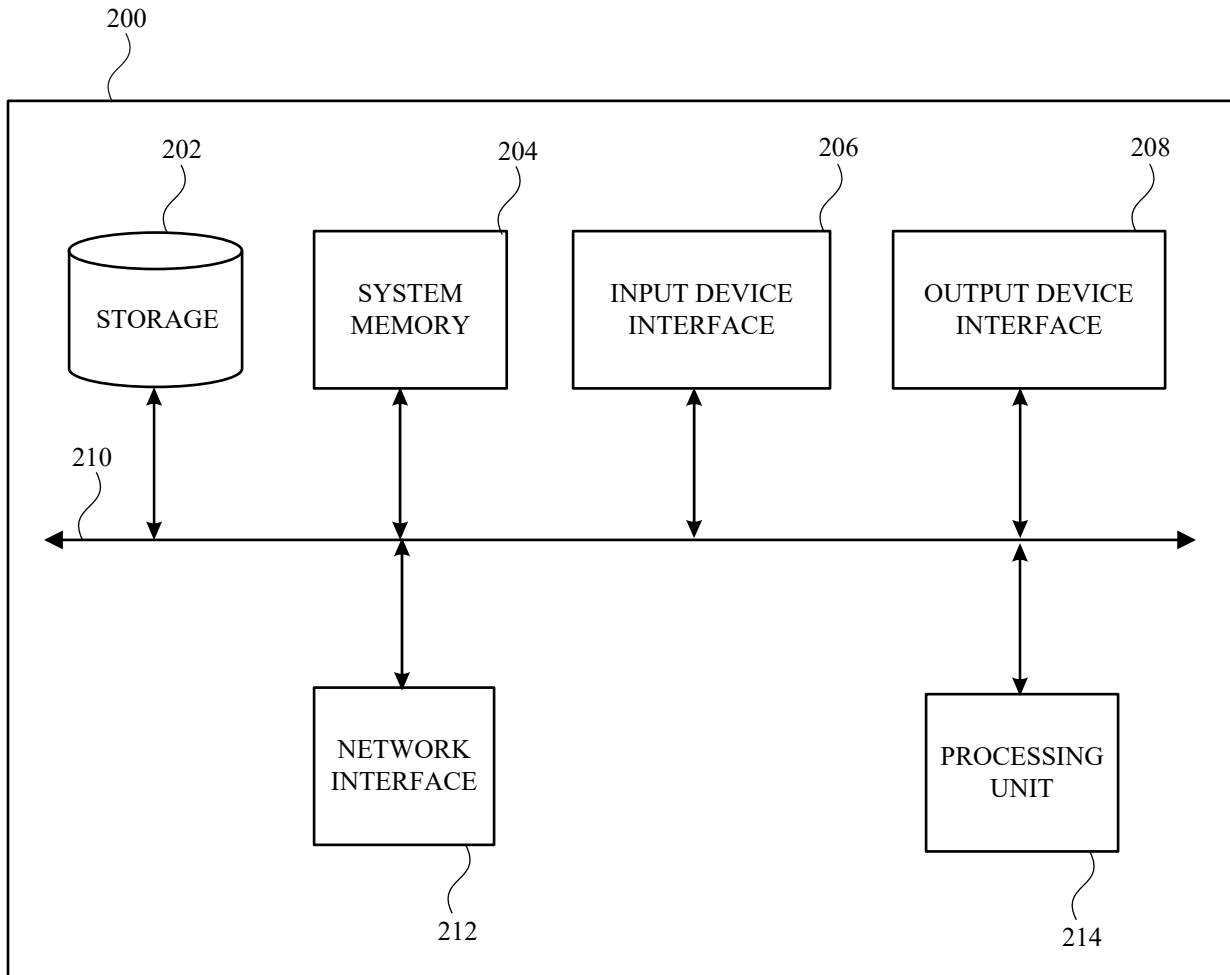


## **ABSTRACT**

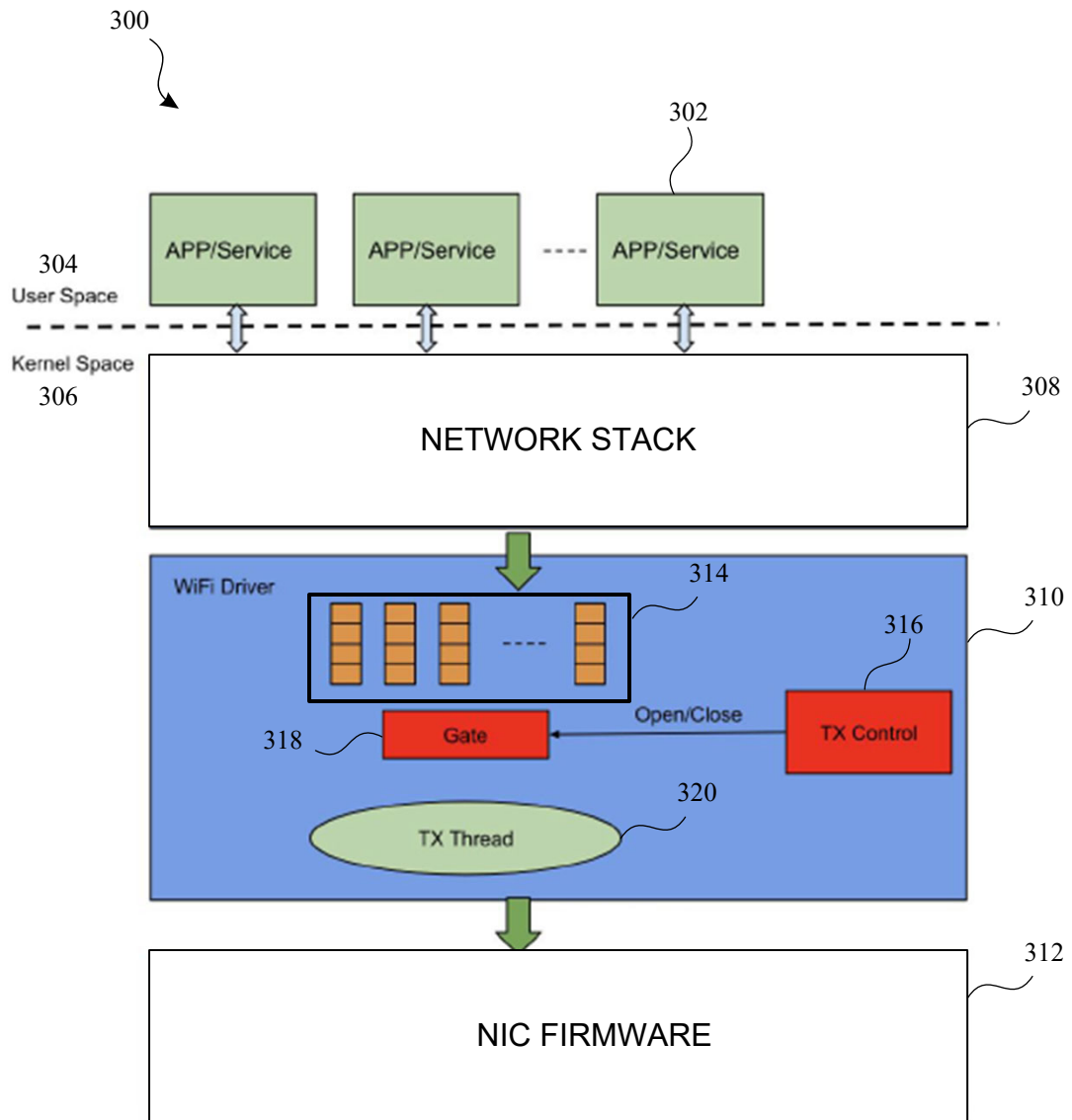
A method for reducing power consumption and thermals of a network interface card by batching network traffic is disclosed. An aspect of the subject technology may include receiving a first control signal, closing a gate corresponding to a buffer in response to the first control signal, receiving one or more data packets with the buffer, receiving a second control signal, opening the gate corresponding to the buffer in response to the second control signal, and transferring one or more data packets of the buffer for transmission in response to the gate opening.



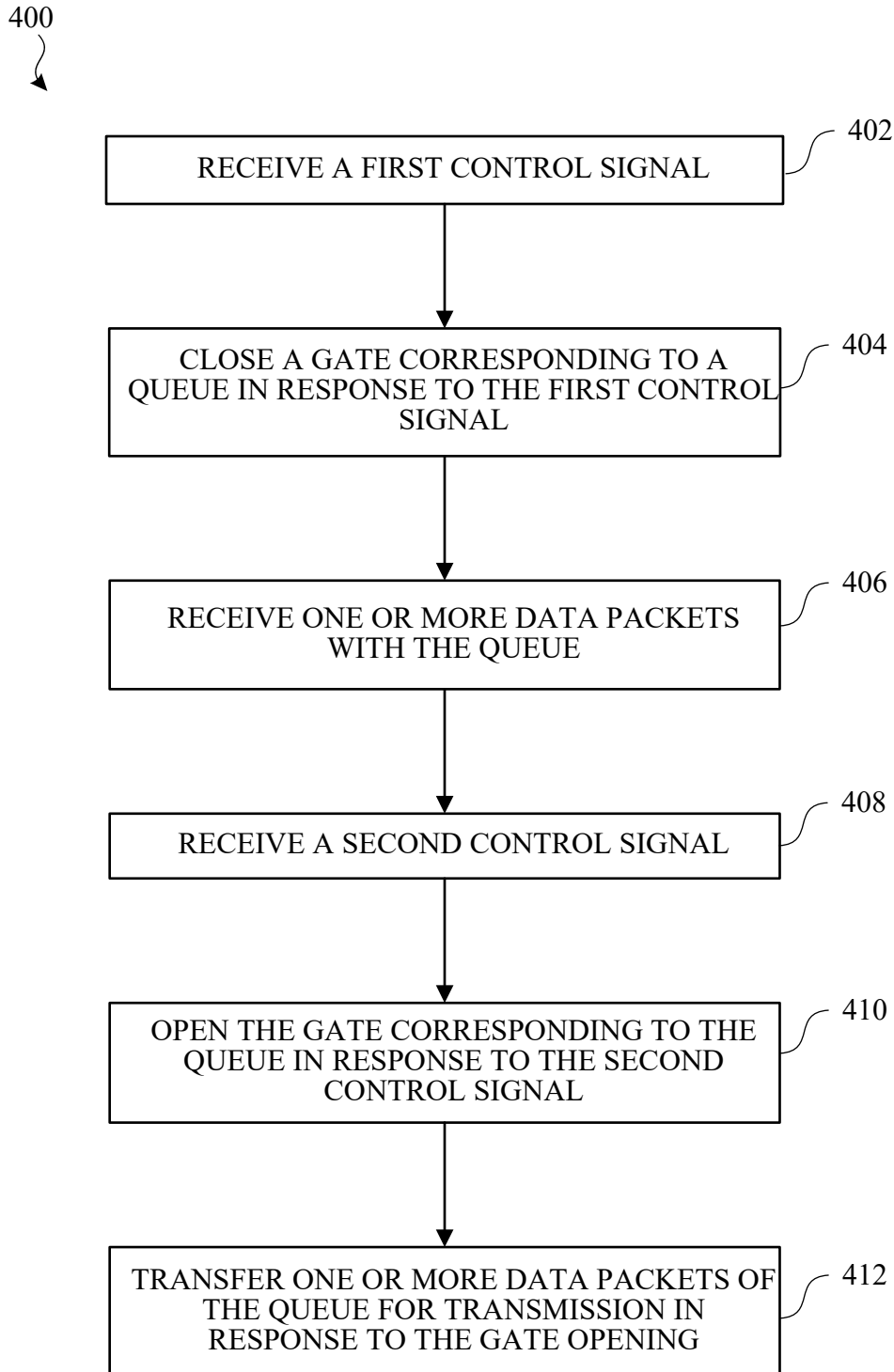
**FIG. 1**



**FIG. 2**



**FIG. 3**



**FIG. 4**

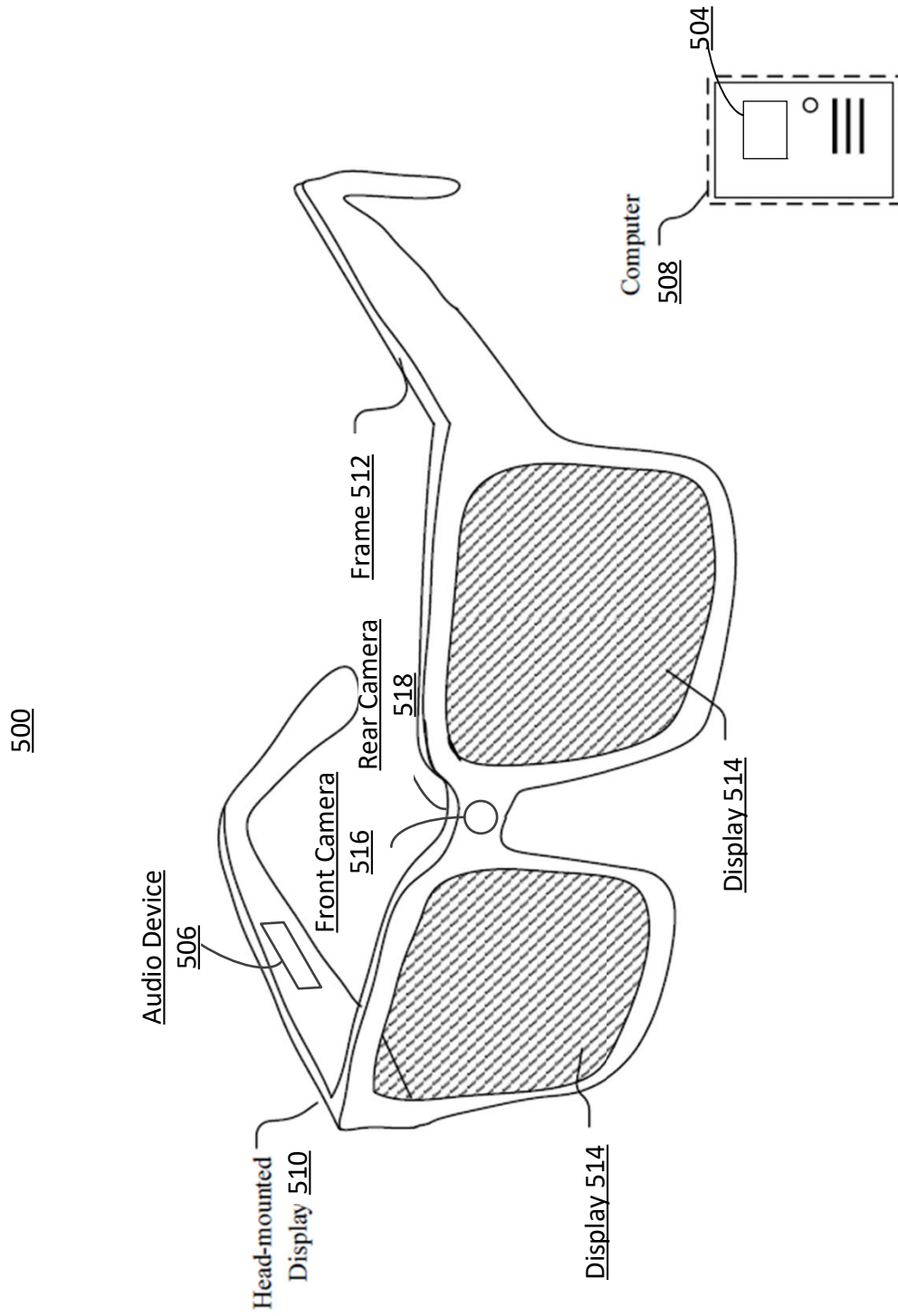


FIG. 5