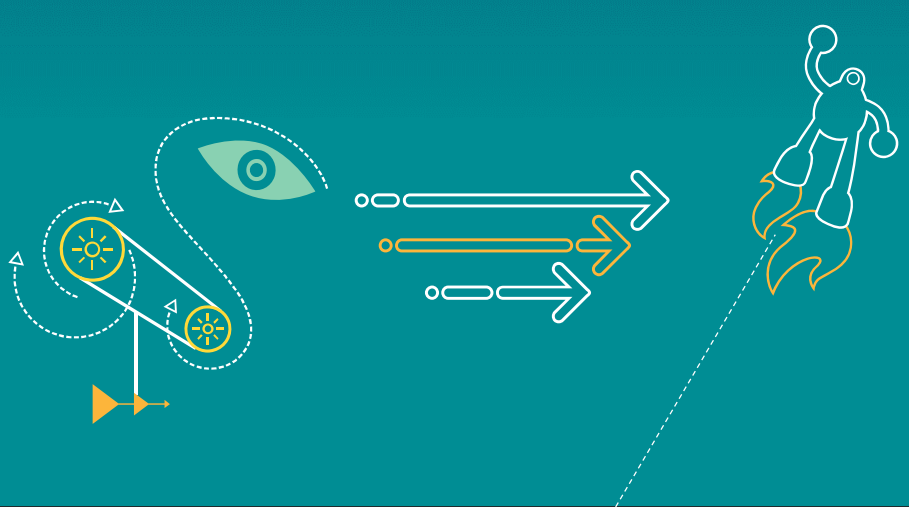
Android N Data Power Manager Overview

Qualcomm Technologies, Inc.



80-P8167-1 A

Confidential and Proprietary – Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

All Qualcomm products referenced herein are products of Qualcomm Technologies, Inc. or its subsidiaries.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

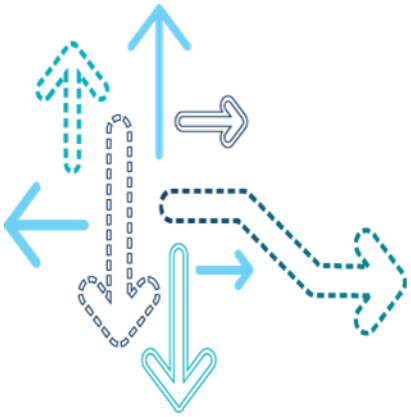
Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision History

Revision	Date	Description
A	September 2016	Initial release

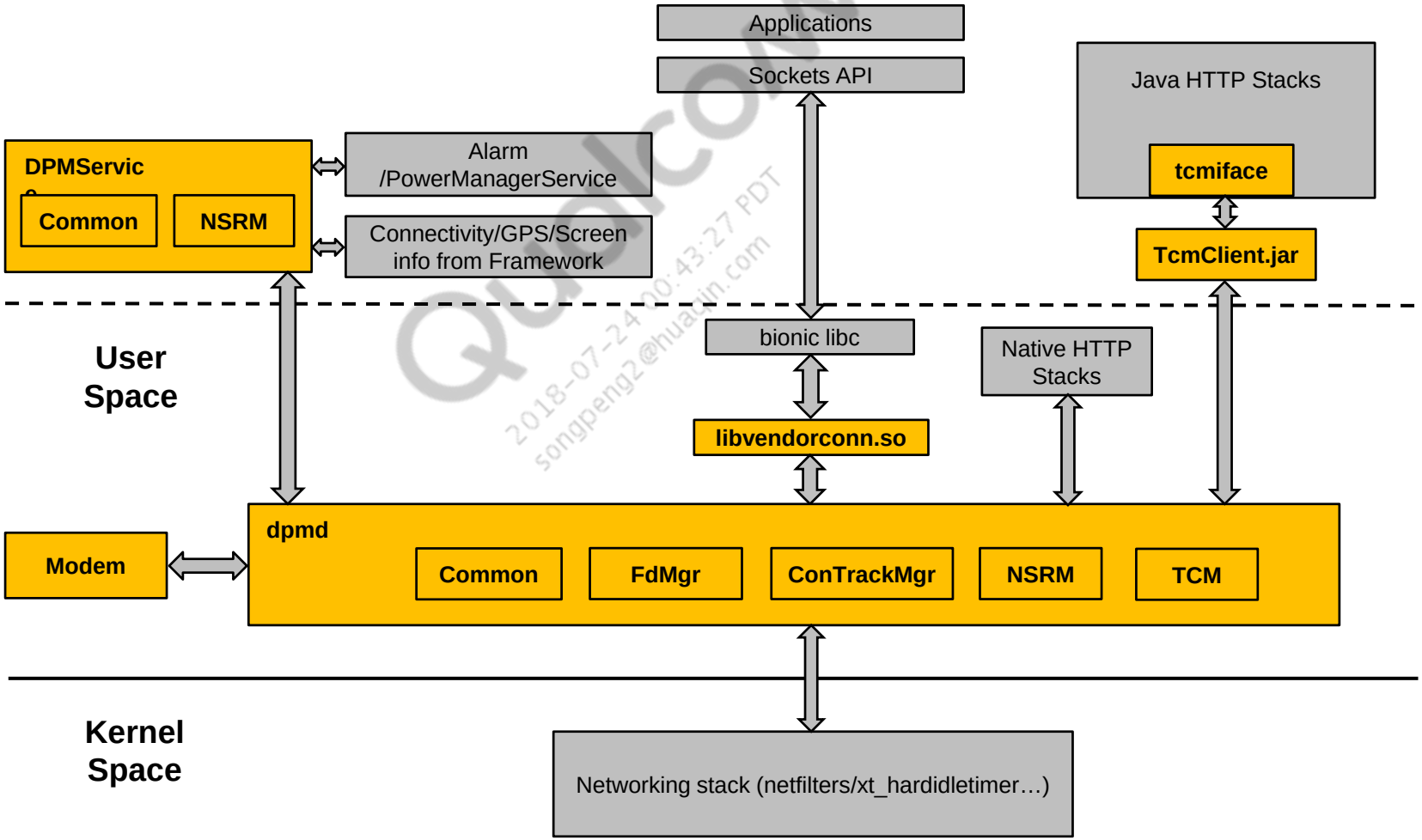
Contents

Introduction

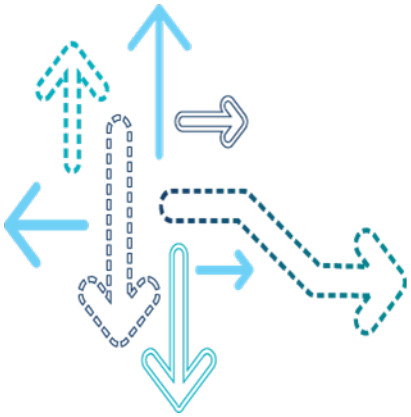


Introduction

- High-Level Architecture



Network Socket Request Manager

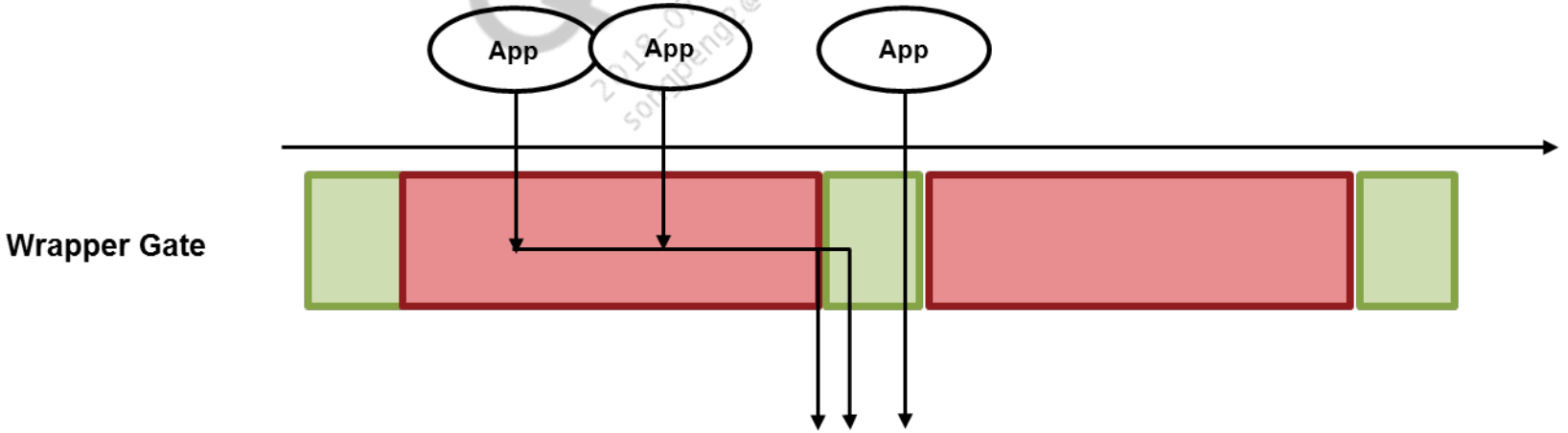


Network Socket Request Manager (NSRM)

- NSRM challenge and solution
  - Issue
    - Applications such as Facebook and Twitter periodically pull data from the network and the requests are asynchronous with respect to time. This results in more RRC connections, i.e., more network signaling, which is directly related to battery consumption on the device
  - Solution
    - Synchronize the socket requests (DNS lookup, connect, or write) from the various applications when the device is in background mode (RRC is dormant, no Wi-Fi, no streaming, etc.)
      - Reduces network signaling
      - Saves device power
      - Has no impact on user experience
    - The algorithm is based on the concept of the NSRM gate

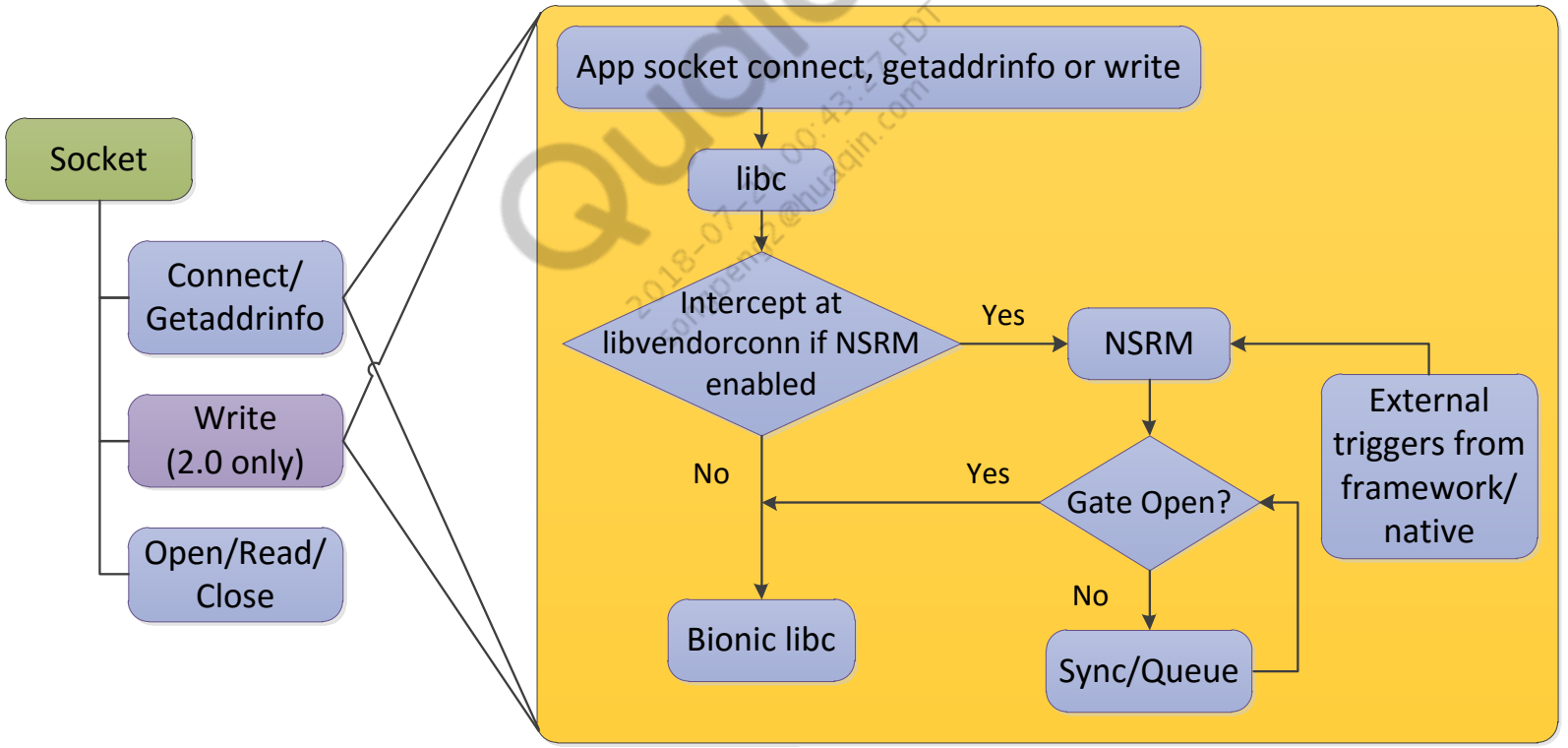
NSRM Gate

- NSRM gate open and close
  - When the gate is closed, selected socket calls are intercepted and synchronized or delayed until the gate is opened.
  - The heart of the algorithm is the mechanism used to decide when to *open* or *close* its gate.
  - Various system states and events affect the gate state.



NSRM 2.0

- NSRM 1.0 vs 2.0
  - NSRM 1.0 synchronizes TCP connect and DNS request
  - In addition, NSRM 2.0 synchronizes TCP write



NSRM 2.0 Timers

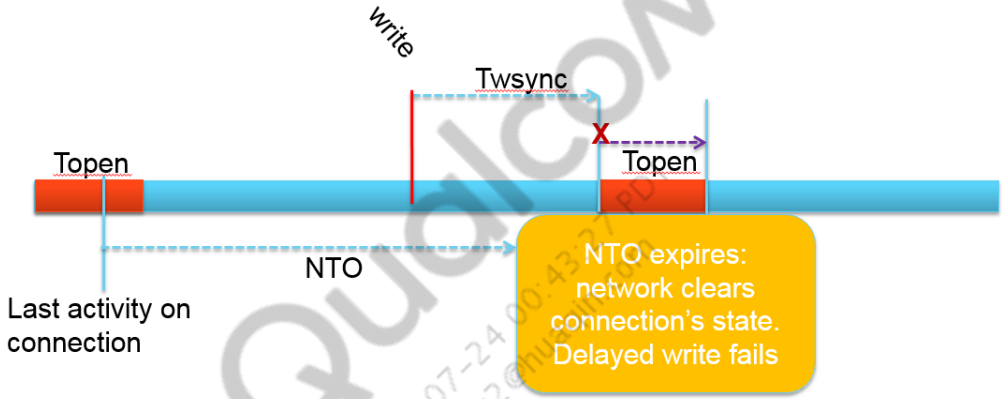
- Tssync, Twsync & Topen timers
  - These timers ensure that the applications are not blocked indefinitely
  - Tssync starts when the first connect or DNS request is captured after the gate is closed
  - Twsync starts when the first write call is captured after the gate is closed
  - The gate opens as soon as Tssync or Twsync expires, unless the gate opens earlier
  - Topen timer starts when the gate state changes to open, and keeps the gate state open until the timer expires. After the timer expires, if all the other conditions are met, the gate state changes to close
  - All these timer values are configurable

NSRM 2.0 Timers (cont.)

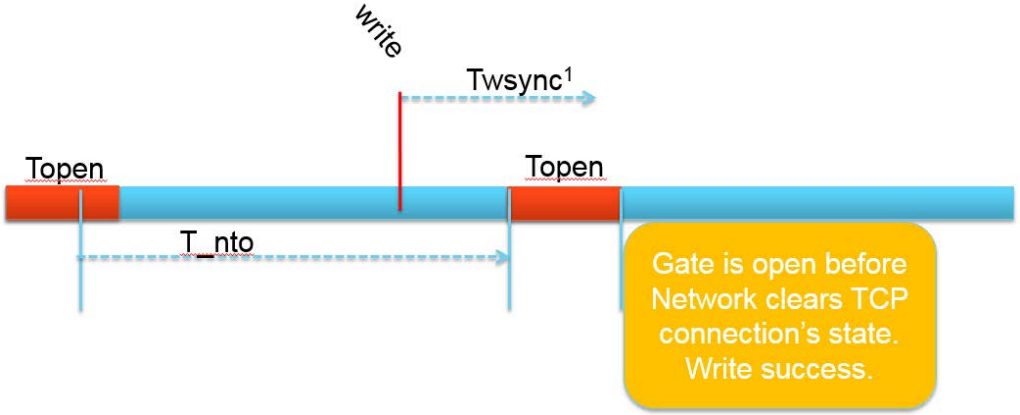
- Tnto timer
  - Tnto ensures that NSRM does not synchronize write traffic beyond the inactive time used by the NAT and Firewalls.
    - Stateful middleboxes in the network, such as NAT or Firewalls, must maintain the state of each connection.
    - The state of the NAT or Firewall is erased after a certain amount of inactive time. When the state is erased, the TCP connection is unusable.
    - Tnto ensures that the synchronized write calls are released before the state is erased.
  - There is one *Tnto timer* for each TCP connection.
  - The gate opens as soon as *Tnto timer* expires, unless the gate opens earlier.
  - *Tnto timer* restarts with its configured value when the TCP connection is used.

NSRM 2.0 Timers (cont.)

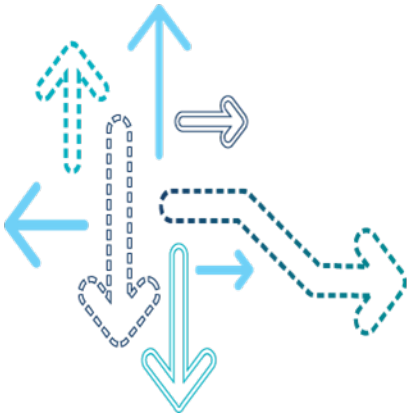
- Tnto example
  - Twsync delaying beyond network inactivity



  - Opening the gate to avoid exceeding NTO



Dynamic NSRM

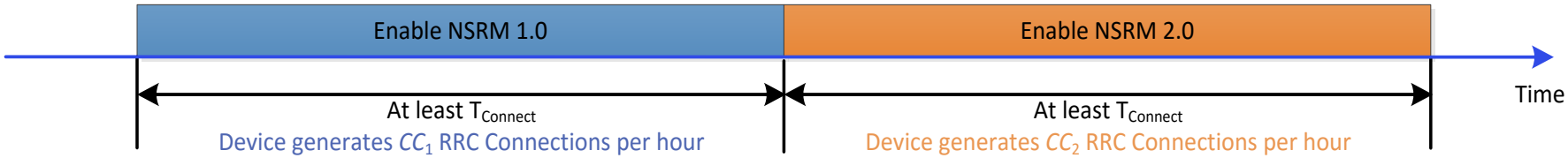


Dynamic NSRM (DNSRM)

- NSRM 2.0 challenge and DNSRM
  - Issue
    - NSRM 2.0 delays write() calls and sometimes causes an issue.
      - Some applications might send more traffic and trigger more RRC connections.
        - For example, in **WhatsApp**, **if write() calls are delayed** for more than 30 seconds, the application **closes the current socket and reconnects to a different server**.
      - NSRM 2.0 should not be applied to this type of application.
    - A search over all applications is not done to find all the NSRM non-friendly applications.
  - Solution
    - Detect dynamically if NSRM is performing well on the UE.
    - Enable NSRM 2.0 dynamically per application.

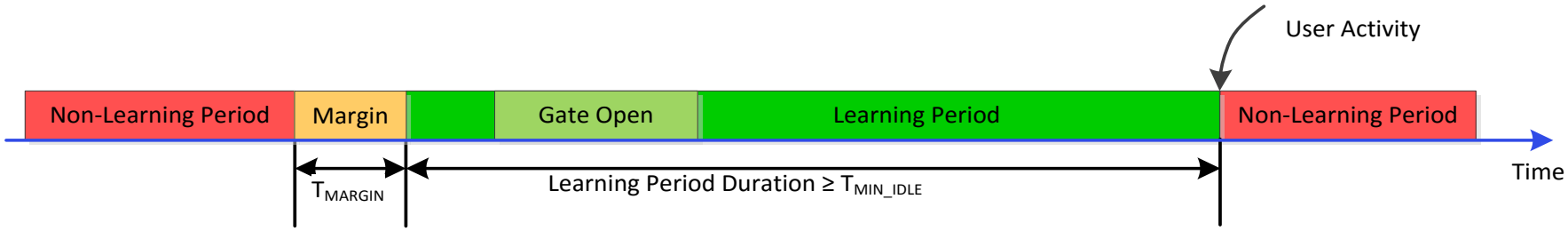
Dynamic NSRM (DNSRM) (cont.)

- Dynamically enable NSRM 2.0
  - Count the number of connect() calls for each application with NSRM 1.0 and with NSRM 2.0. Count for at least $T_{Connect}$ (240) minutes
    1. If $CC_2 > \boldsymbol{BAD_{Thr}} \times CC_1$, disable NSRM 2.0
    2. Else if $CC_2 \leq \boldsymbol{GOOD_{Thr}} \times CC_1$, enable NSRM 2.0
    3. Else recheck
  - Only count when the user is not using the device
  - Rerun the algorithm when one of the following occurs:
    - The application is updated
    - It is 30 days since the last check

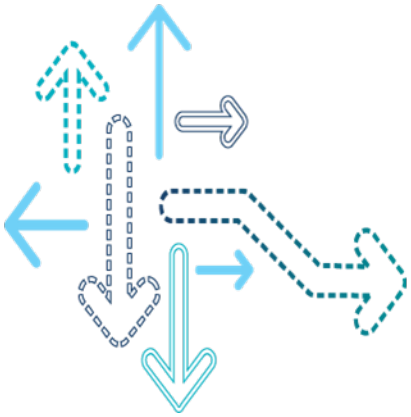


Dynamic NSRM (DNSRM) (cont.)

- Learning period in device idle state
  - Do *not* count when the user is using the device. This is the non-learning period.
  - Count only when the device is idle. This is the learning period.
    - Add a margin of T_{MARGIN} (60) seconds after the end of the non-learning period.
      - Allows the applications to cool down.
    - The learning period must be at least T_{MIN_IDLE} (5) minutes.
      - There might be no RRC connections during a short idle time, which might make the counting inaccurate.
  - The application learning period is when the application is running and during a valid period.



Doze Feature



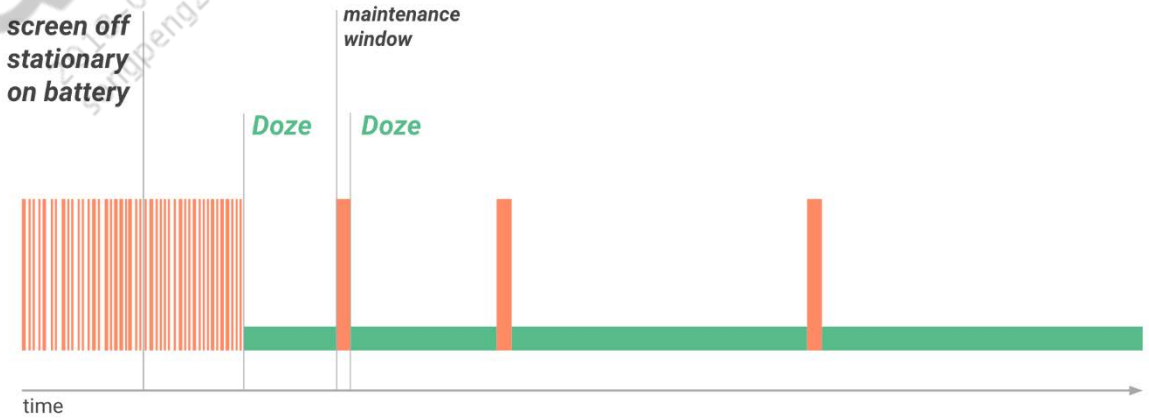
Doze Feature

- What is Doze feature
  - Doze feature prevents the battery from draining
    - When the device is unplugged with the screen off for about 30 minutes, it shifts into Doze mode
  - While Doze is active
    - No network access
    - Ignore “wakelocks” when apps try to keep the device from going to sleep
    - No background tasks allowed
    - Alarm/sync deferred

Doze Modes

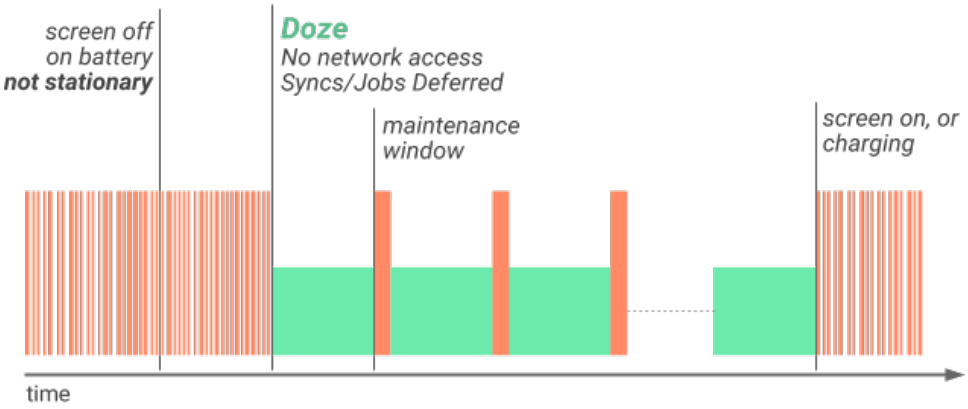
- Doze deep mode
  - Doze mode is available from Marshmallow
  - Waits for N minutes and starts motion detection algorithm to check if device is idle stationary long enough
    - If the device is in idle stationary, Doze deep mode goes to STATE_IDLE
    - Else, goes to STATE_INACTIVE and keeps cycling STATE_IDLE_PENDING, STATE_SENSING and STATE_LOCATING states depending on device motion or stationary

- No wakelock allowed
- Alarms deferred to next maintenance window
- No GPS/wifi scan allowed



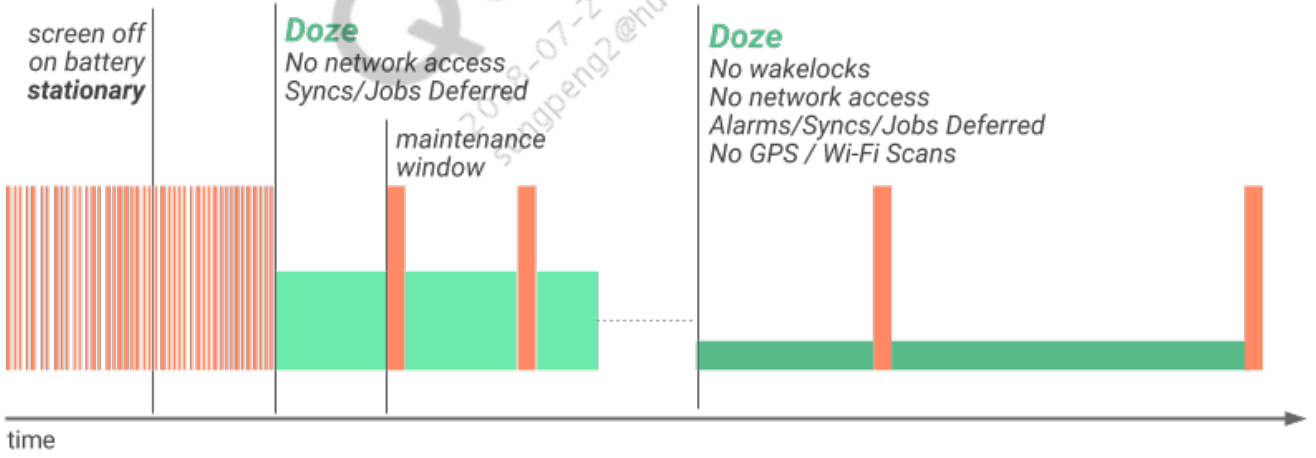
Doze Modes (cont.)

- Doze light mode
  - Doze light is an extension to the Doze feature in Nougat
  - Screen off and charging off means LIGHT_STATE_INACTIVE
  - Waits for N minutes and checks system activity if any current alarm, wakelock, or network activity
    - If there is any pending system activity, Doze light mode goes to LIGHT_STATE_PRE_IDLE and starts pre idle timer with LIGHT_PRE_IDLE_TIMEOUT, and moves to LIGHT_STATE_IDLE when the timer has expired
    - Else, goes to LIGHT_STATE_IDLE
  - Wakelocks and alarms are available



Doze Modes (cont.)

- Doze light mode and deep mode
  - Both doze light and deep modes work independent of each other, except when deep mode is in STATE_IDLE_MAINTENANCE then light mode is in LIGHT_STATE_OVERRIDE



Doze Mode States

Doze Light Mode States

State	Description
LIGHT_STATE_ACTIVE	Device is currently active
LIGHT_STATE_INACTIVE	Screen off, Charging off and waiting to move into first light idle
LIGHT_STATE_PRE_IDLE	Device is about to go idle for first time , wait for current work to complete
LIGHT_STATE_IDLE	Device in light idle state , trying to stay asleep as much possible
LIGHT_STATE_IDLE_ MAINTENANCE	Device is in light idle state but in regular maintenance mode
LIGHT_STATE_OVERRIDE	Light idle state is overridden , now moving to deep doze state

Doze Deep Mode States

State	Description
STATE_ACTIVE	Device is currently active
STATE_INACTIVE	Device is inactive (screen off, no motion) and we are waiting to for idle
STATE_IDLE_PENDING	Device is past the initial inactive period, and waiting for the next idle period
STATE_SENSING	Device is currently sensing motion
STATE_LOCATING	Device is currently finding location (and may still be sensing)
STATE_IDLE	Device is in the idle state, trying to stay asleep as much as possible
STATE_IDLE_MAINTENANCE	Device is in the idle state, but temporarily out of idle to do regular maintenance

Doze Mode with NSRM

- NSRM coexistence with Doze

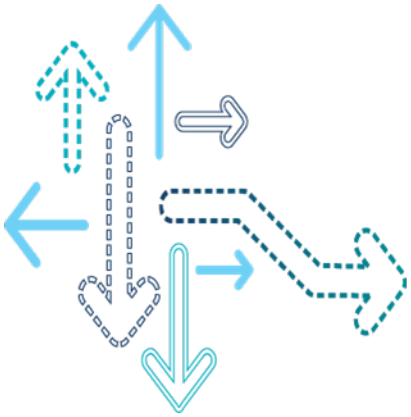
	Doze Disabled	Doze Enabled
NSRM Disabled	No Power Management	Doze Power Management
NSRM Enabled	NSRM Power Management	Configure NSRM gate opened when Doze Light is in LIGHT_STATE_IDLE or LIGHT_STATE_IDLE_MAINTENANCE and Doze Deep is in STATE_IDLE or STATE_IDLE_MAINTENANCE => Vendor Trigger State to 1 (Enabled) NSRM works as usual in all other Doze states => Vendor Trigger State to 0 (Disabled)

Logging for DPM

- Required logs with level
  - adb logcat, kernel, and tcpdump commands
    - adb logcat -b main -b radio -b system -b events -v threadtime | tee log.log
    - adb shell cat /proc/kmsg | tee kmsg.log
    - adb shell tcpdump -nvi rmnet0 -s 0 -w /sdcard/tcpdump.pcap
  - DPM log messages should be captured in QXDM log with enabling **DPM [10400 ~ 10414] – all level of messages**

persist.dpm.loglevel	dpmd logs	java logs
Not set	QXDM(E+W+I)	ADB(E+W+I)
3974	QXDM(E+W+I+D)	Complete logs
7825	QXDM(E+W+I+D+V)	Complete logs

Reference Logs



Reference Logs

- NSRM gate state change and socket synchronization

```
03-20 03:51:01.216  283  283 V DPM  : |NSRM:GATESM| void DpmNsrmGateState::HdmiStateInd(NsrmHDMIStateEnum_t):592 Ind:HDMI is connected.
03-20 03:51:01.216  283  283 V DPM  : |NSRM:GATESM| void DpmNsrmGateState::TransitionState():877 Current State: Gate is closed., Event Mask: 0x30a Event Enable Mask: 0xd00
03-20 03:51:01.216  283  283 V DPM  : |NSRM:GATESM| New State: Gate is open.
03-20 03:51:01.216  283  283 V DPM  : |NSRM| Release 4 number of uids.
03-20 03:51:01.216  283  283 V DPM  : |NSRM| Release 1 number of sockets belonging to uid 10006
03-20 03:51:01.216  283  283 V DPM  : |NSRM| Release socket 3 belonging to uid 10006
03-20 03:51:01.217  283  283 V DPM  : |NSRM| Release 2 number of sockets belonging to uid 10008
03-20 03:51:01.217  283  283 D DPM  : |NSRM:GATESM| Topen timer of 10 seconds has started.
03-20 03:51:01.217  283  283 V DPM  : |NSRM:GATESM| void DpmNsrmGateState::TOpenTimerInd(NsrmTopenTimerStateEnum_t):782 Ind:Topen timer is started.
03-20 03:51:11.219  283  283 V DPM  : |NSRM:GATESM| void DpmNsrmGateState::TOpenTimerInd(NsrmTopenTimerStateEnum_t):782 Ind:Topen timer is expired.
03-20 03:51:42.671  283  283 V DPM  : |NSRM:GATESM| void DpmNsrmGateState::HdmiStateInd(NsrmHDMIStateEnum_t):592 Ind:HDMI is disconnected.
03-20 03:51:42.671  283  283 V DPM  : |NSRM:GATESM| New State: Gate is closed.
03-20 03:51:46.746  283  283 V DPM  : |NSRM| NsrmSockReleasePermitType_t DpmNsrmSOI::Synchronize(int, NsrmSocketClassType, unsigned int*, int):401 socketType 0
03-20 03:51:52.181  283  283 D DPM  : |NSRM| com.android.browser appname wasnt found.
03-20 03:51:52.181  283  283 V DPM  : |NSRM| No appname found. Mode:  exclusion.
03-20 03:51:52.181  283  283 V DPM  : |NSRM| Synchronize: Write socket is present in queue
03-20 03:51:52.182  283  283 V DPM  : |NSRM| Total number of unique uids queued: 2
03-20 03:51:52.182  283  283 D DPM  : |NSRM:GATESM| void DpmNsrmGateState::startTSyncTimer(NsrmTimerClassType, int):261
03-20 03:51:52.182  283  283 D DPM  : |NSRM:GATESM| getExpectedNToExpiry: lastDataActivityTime = 27 seconds, ntoValue = 300 seconds NTO_TIME_MARGIN = 10
03-20 03:51:52.182  283  283 V DPM  : |NSRM:GATESM| Started Twsync timer id 5, actual delay 247591 ms
03-20 03:52:15.194  283  283 D DPM  : |NSRM| com.android.vending appname wasnt found.
03-20 03:52:15.194  283  283 D DPM  : |NSRM| com.android.vending appname wasnt found.
03-20 03:52:15.194  283  283 V DPM  : |NSRM| DpmRetType DpmNsrmApplication::AddSocket(unsigned int):69
```

Reference Logs (cont.)

- NSRM gate state change with Doze deep

08-22 05:50:18.330 2447 2447 V DPMJ : |DPM:NSRM| recieved idle intent: android.os.action.**DEVICE_IDLE_MODE_CHANGED**
08-22 05:50:18.331 2447 2447 I DPMJ : |DPM:NSRM| updateDozeTriggerStatus
08-22 05:50:18.344 2447 2447 I DPMJ : |DPM:NSRM| doze deep detailed **state: 0**light state 0
08-22 05:50:18.345 2447 2447 I DPMJ : |SERVICE| disable Trigger 1 id 1
08-22 05:50:18.350 2447 2447 I DPMJ : |SERVICE| disableVendorTrigger bDpmdCmd == true && mVendorTriggerState == true
08-22 05:50:18.351 2447 2447 I DPMJ : |SERVICE| sendVendorTriggerState: state: false
08-22 05:50:18.351 2447 2447 I DPMJ : |SERVICE| disableVendorTrigger id 1 mVendorTriggerInfo.state false state false
08-22 05:50:18.351 901 901 V DPM : |COMMON| processing DPM event 'DPM_S_NOTIFY_VENDOR_TRIGGER_STATE_CHG' (16) [token 48, count 48]
08-22 05:50:18.351 901 901 V DPM : |COMMON:COM| data: 1 ints
08-22 05:50:18.351 901 901 I DPM : |COMMON:DSM| DSM processCommand: Rcvd command, cmdType=16
08-22 05:50:18.351 901 901 I DPM : |COMMON:DSM| DSM processCommand: Nsrm Vendor Event, state:0
08-22 05:50:18.351 901 901 I DPM : |COMMON:DSM| handleVendorTriggerEvent: vendor trigger State Event, newstate:0 oldstate:1
08-22 05:50:18.351 901 901 V DPM : |NSRM:TRG| static void DpmNsrmBackgroundEvtHdlr::dsmEventHdlr(DpmDsmEvent, const void *, void *):208
08-22 05:50:18.351 901 901 D DPM : |NSRM:TRG| BackgroundStateChgEvtHandler Event 14 occurred
08-22 05:50:18.352 901 901 V DPM : |NSRM:TRG| DPM_DSM_NSRM_VENDOR_TRIGGER_STATE_CHANGE_EVENT handled.
08-22 05:50:18.352 901 901 V DPM : |NSRM:TRG| void DpmNsrmBackgroundEvtHdlr::BackgroundStateChgEvtHdlr_(const DpmDsmBackgroundEventData *):312
08-22 05:50:18.352 901 901 D DPM : |NSRM:TRG| BackgroundStateChgEvtHdlr_ event 12 on 0
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| void DpmNsrmState::VendorTriggerStateInd(NsrmVendorTriggerStateEnum_t):258 Ind:Vendor Trigger is disabled.
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| virtual void DpmNsrmGateState::TransitionState():426 Current State: Gate is open., Event Mask: 0x208 Event Enable Mask: 0x4c00
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| RRC State(valid only if wwan connected): 0 (0:not connected; 1:connected)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Screen State: 0 (0:Off; 1:On)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| GPS State: 0 (0:Stopped; 1:Started)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| USB State: 1 (0:Disconnected; 1:Connected)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Headset State: 0 (0:Disconnected; 1:Connected)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Bluetooth State: 0 (0:Disconnected; 1:Connected)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Music State: 0 (0:Inactive; 1:Active)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Microphone State: 0 (0:Mute; 1:On)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| HDMI State: 0 (0:Disconnected ; 1:Connected)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| WLAN Connectivity State: 1 (0:Disconnected; 1:Connected)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Tsync timer State: 0 (0:On ; 1:Expired)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Topen timer State: 0 (0:Expired ; 1:On)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Speaker State: 0 (0:Off ; 1:On)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| Emergency Alert State: 0 (0:Off ; 1:On)
08-22 05:50:18.352 901 901 V DPM : |NSRM:GATESM| **Vendor Trigger State: 0 (0:Disabled ; 1:Enabled)**
08-22 05:50:19.667 901 901 V DPM : |NSRM:GATESM| New State: Gate is closed.

Reference Logs (cont.)

- NSRM gate state change with Doze light

08-22 05:34:45.545 2447 2447 V DPMJ : |DPM:NSRM| recieved idleLight Intentandroid.os.action.**LIGHT_DEVICE_IDLE_MODE_CHANGED**
08-22 05:34:45.545 2447 2447 I DPMJ : |DPM:NSRM| updateDozeTriggerStatus
08-22 05:34:45.546 2447 2447 I DPMJ : |DPM:NSRM| doze deep detailed state: 1**light state 4**
08-22 05:34:45.547 2447 2447 E DpmApi : enableTrigger
08-22 05:34:45.547 2447 2447 E DpmApi : enableTriggerMethod
08-22 05:34:45.547 2447 2447 I DPMJ : |SERVICE| enable Trigger 1 id 1
08-22 05:34:45.547 2447 2447 I DPMJ : |SERVICE| enableVendorTrigger mVendorTriggerState == false
08-22 05:34:45.547 2447 2447 I DPMJ : |SERVICE| sendVendorTriggerState: state: true
08-22 05:34:45.548 2447 2447 I DPMJ : |SERVICE| enableVendorTrigger id 1 vendorInfo.state true state true
08-22 05:34:45.548 901 901 V DPM : |COMMON| processing DPM event 'DPM_S_NOTIFY_VENDOR_TRIGGER_STATE_CHG' (16) [token 43, count 43]
08-22 05:34:45.549 901 901 I DPM : |COMMON:DSM| DSM processCommand: Nsrm Vendor Event, state:1
08-22 05:34:45.549 901 901 I DPM : |COMMON:DSM| handleVendorTriggerEvent: vendor trigger State Event, newstate:1 oldstate:0
08-22 05:34:45.549 901 901 V DPM : |NSRM:TRG| static void DpmNsrmBackgroundEvtHdlr::dsmEventHdlr(DpmDsmEvent, const void *, void *):208
08-22 05:34:45.549 901 901 D DPM : |NSRM:TRG| BackgroundStateChgEvtHandler Event 14 occurred
08-22 05:34:45.549 901 901 V DPM : |NSRM:TRG| DPM_DSM_NSRM_VENDOR_TRIGGER_STATE_CHANGE_EVENT handled.
08-22 05:34:45.549 901 901 V DPM : |NSRM:TRG| void DpmNsrmBackgroundEvtHdlr::BackgroundStateChgEvtHdlr_(const DpmDsmBackgroundEventData *):312
08-22 05:34:45.549 901 901 D DPM : |NSRM:TRG| BackgroundStateChgEvtHdlr_ event 12 on 1
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| void DpmNsrmState::VendorTriggerStateInd(NsrmVendorTriggerStateEnum_t):258 Ind:Vendor Trigger is enabled.
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| virtual void DpmNsrmGateState::TransitionState():426 Current State: Gate is closed., Event Mask: 0x4208 Event Enable Mask: 0x4c00
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| RRC State(valid only if wwan connected): 0 (0:not connected; 1:connected)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Screen State: 0 (0:Off; 1:On)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| GPS State: 0 (0:Stopped; 1:Started)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| USB State: 1 (0:Disconnected; 1:Connected)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Headset State: 0 (0:Disconnected; 1:Connected)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Bluetooth State: 0 (0:Disconnected; 1:Connected)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Music State: 0 (0:Inactive; 1:Active)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Microphone State: 0 (0:Mute; 1:On)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| HDMI State: 0 (0:Disconnected ; 1:Connected)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| WLAN Connectivity State: 1 (0:Disconnected; 1:Connected)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Tsync timer State: 0 (0:On ; 1:Expired)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Topen timer State: 0 (0:Expired ; 1:On)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Speaker State: 0 (0:Off ; 1:On)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| Emergency Alert State: 0 (0:Off ; 1:On)
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| **Vendor Trigger State: 1 (0:Disabled ; 1:Enabled)**
08-22 05:34:45.549 901 901 V DPM : |NSRM:GATESM| New State: Gate is open.
08-22 05:34:45.549 901 901 V DPM : |NSRM| void DpmNsrmSOI::ReleaseAllConnections():1171

Questions?

https://createpoint.qti.qualcomm.com