

# BREW<sup>®</sup> Network Programming

Rob Walker, Principal Engineer

*QUALCOMM Incorporated*



# BREW Networking

- **Real TCP/IP *Internet* access**
  - TCP: reliable bi-directional byte streams
  - UDP: connectionless packet delivery
- **Limited number of sockets (varies by handset)**
  - Minimum is 1 UDP and 1 TCP socket, but consult device data sheets

# Capabilities: CDMA

- **Similar to desktop demand-dial PPP without the modem delay**
  - Typically dynamically-assigned IP addresses
  - BREW handles connecting/disconnecting
  - “Always on” as in ready for outbound connections
  - Most networks allow long-lived idle connections without consuming air resources

- **Some of the critical peak download speeds:**

IS-95A	14.4Kbit/s
1xRTT	153Kbit/s
1xEV-DO	2.4Mbit/s
1xEV-DO RevA	3.1Mbit/s

## Capabilities: Networks

- **Operator may employ Network Address Translation (NAT)**
  - Handsets assigned addresses from a non-routable (private) pool; address translated in gateway
  - Outbound (phone-initiated) TCP and UDP is generally unimpaired
  - Inbound fails when peer is outside the private network
- **Some networks may not support direct handset-to-handset connections**

## Capabilities: Networks (cont.)

- **Networks vary, but general rule is:**
  - Use outbound connections (HTTP, POP3, IMAP, etc.)
  - IP-based asynchronous notification is problematic (current best practice is BREW-directed SMS)
  - Peer-to-peer applications should employ a rendezvous server for robustness
- **Some networks may support permanent IP addresses via MobileIP (or even IPv6)**

# API Overview

- **Modeled after BSD “sockets” API**
- **A socket is a TCP or UDP endpoint**
- **Some Internet-specific simplifications**
- **Non-blocking interface only**
- **Callback-based notifications**
- **INetMgr and ISocket interfaces expose networking**

# INetMgr Interface

- **INetMgr obtained via:**
  - `ISHELL_CreateInstance(piShell, AEECLSID_NET, &piNet)`
- **Represents a data network**
  - PPP state (OPEN, ASLEEP, CLOSED)
  - Single IPv4 address
- **Parent of all ISocket instances**
- **Often used to access other functions and then released**

# INetMgr Methods

- **OpenSocket()**
  - Creates a STREAM or DGRAM ISocket
- **GetHostByName()**
  - Name Service (resolver)
    - Retrieves “A” record via IDNS
    - Caches results for performance
- **NetStatus()**
  - Retrieve instantaneous network state
- **OnEvent()**
  - Register callback for asynchronous network state changes

## INetMgr Methods (2)

- **GetMyIPAddr()**
  - Local IP address
    - Zero if PPP down
- **SetLinger()**
  - Linger timer
    - Defaults to 30 seconds
    - Max of all INetMgr instances
    - Close PPP *linger* seconds after last socket activity
- **SetOpt() / GetOpt()**
  - Generic Control
    - RLP3 settings
    - PPP credentials
    - DNS servers

# ISocket Interface

- **Represents an IPv4 socket**
  - **STREAM** or **DGRAM**
- **INetMgr manages network on behalf of sockets**
  - **Opens PPP** only when needed
  - **Local IP address** needed for **Bind()**, **Connect()**, or **SendTo()**
- **Loopback support**
  - **INADDR\_BREW\_LOOPBACK** handled entirely by BREW
  - **INADDR\_LOOPBACK**

# ISocket Methods

- **Bind() / GetSockName()**
  - Request / retrieve local socket address
- **Listen() / Accept()**
  - TCP server sockets
- **Connect(), GetPeerName()**
  - Request, retrieve remote TCP socket address
- **Read(), ReadV(), Write(), WriteV()**
  - TCP data stream
- **Shutdown(), Close()**
  - Graceful TCP termination

## ISocket Methods (2)

- **SendTo(), RecvFrom()**
  - UDP datagrams
- **GetLastError()**
  - Specific error code of last operation (e.g. ECONNRESET)
- **Readable(), Writeable(), Cancel()**
  - Non-blocking I/O
- **SetOpt() / GetOpt()**
  - Socket options (TCP keep-alive, receive buffer size, etc.)
- **IOctl()**
  - Generic control, extensions

# Non-blocking I/O: Retry Functions

- **Read(), Write() and related functions**
  - Return immediately, indicating success, failure, or “try again”
  - These do not do any work after returning or remember pointer passed
  - These can be called anytime. Timers are an inefficient solution. Use Readable() and Writable()
- **Use Readable() & Writable() to wait for retries**
  - Notified when progress may be made
  - “Progress” may be a failure case.

# IDNS Interface

- **High level abstraction for complex DNS transactions**
  - Indirectly used by `INETMGR_GetHostByName()`
  - Not typically used directly by applications
- **AddQuestion()**
  - Multiple questions can be added to a single query
    - Multiple domain names
    - Multiple record types (e.g. A, MX, SVR)
- **Start()**
  - Performs query and notifies application when response is ready
- **GetResponse()**
  - Response contains data for each answer
    - Authoritative Answer?
    - TTL, etc.
- **ParseDomain()**
  - Extracts a (compressed) domain name from the response record

# Advanced Features

- **Multiple networks (INETMGR\_SelectNetwork())**
  - INETMGR\_SelectNetwork()
  - Default data network (e.g. CDMA Service Network)
  - Multiple PDP profiles in UMTS
  - EVDO (BCMCS) and MediaFLO™ Broadcast channels
- **IDNSConfig**
  - Accessed via IDNS\_QueryInterface()
  - Used to override default behavior
  - SelectNetwork(), GetServers(), SetServers()
- **Multicast socket options**
  - Join / Leave a multicast group

## Advanced Features (2)

- **Dormancy timer separate from linger timer**
  - 3.x differs from 2.x: `INETMGR_SetDormancyTimeout()`
- **Quality of Service (QoS)**
  - Modeled as separate networks
- **IWeb**
  - HTTP/1.1, “transfer-coding: chunked” allows posting of streamed content (HTTP/1.0 still available via a flag)
  - IPv6 support
  - DNS-based failover for servers with multiple IPs

## Advanced Features (3)

- **ISockPort**

- Next generation socket API (ISocket deprecated)
- Generic address structure (not restricted to IPv4)

- **INetwork (as of BREW 3.1.5)**

- Next generation network API (INetMgr deprecated)
- Generic address structure (not restricted to IPv4)

- **IPv6 support (as of BREW 3.1.5)**

- Impacts IWeb, IWebUtils, INetwork, IDNS, and ISockPort

# Performance

- **Device dependent**
  - CPU speed
  - TCP buffer sizes
- **Technology dependent**
  - Various flavors of CDMA have different theoretical max
  - Overhead for TCP/IP, PPP, and RLP
  - Poor signal strength will degrade performance
- **Web typical download speeds (vs. theoretical)**
  - MSM5100<sup>®</sup> IS-95A might see only 8kbps (14.4k)
  - MSM6100<sup>®</sup> 1x could experience 80kbps (153k)
  - MSM6500<sup>®</sup> EV-DO can approach 1Mbps (2.4M)

# Questions?