A walkthrough of the TinyOS BLIP stack for UDP transmission, forwarding, and reception

Network Layer

Transport Layer

Host-to-network layer

App sends UDP packet

DdpP.UDP.sendtos
  Add payload IOV

DdpP.UDP.sendtosv
  Prefix UDP header IOV

IPProtocolsP.IP.Send
  Prefix IP header IOV, set TTL, protocol type

IPForwardingEngineP.IP.Send
  Lookup next hop in routing table

IPForwardingEngineP.IPDecodeSend
  IID Pool?

IPNeighborDiscoveryP.IPForward
  Resolve local addr

IPDispatchP.IPLower
  Check if free frag available
  - If not, ERETRY
  - Clear fragment (CC2420TinyOSNetworkP.BarePacket.clear)

CC2420TinyOSNetworkP.BareSend
  Create unique DSN in packet header

PacketLinkP.Send
  Retry until success or num_retries

CC2420CanP.Send
  Talk to hw/driver

Hardware transmits

App receives UDP packet

DdpP.UDP.recvfrom

DdpP.IP.recv
  Verify UDP hdr checksum

Check port and see if anyone has called bind() for this port

IPProtocolsP.IP.recv
  Check protocol type UDP/TCP/ICMP

IPProtocolsP.SubIP.recv
  Check that IP header is valid

IPForwardingEngineP.IPrecv
  Signal Raw packet if anyone is interested
  - Check if it is destined for local address

IPNeighborDiscoveryP.IPLower.recv
  Check if pkt is for me
  - Forwarding
  - SubReceive

IPDispatchP.IPLower.recv
  Unpack headers into IOV and de-frag if needed into "recon" buffer

Srikanth Nori
Autonomous Networks Research Group
University of Southern California