Network Interrupt Service Routine

Robert N. M. Watson

16 May 2007

FreeBSD Developer Summit
BSDCan 2007

UNIVERSITY OF
CAMBRIDGE

netisr

- Historically
  - Could not access structures from interrupt handler
  - Asynchronous execution for network processing
  - Piggy-backed on interrupt handling facilities (ISR)
- Today
  - Used to avoid recursion, re-entrance, deadlock
  - Direct and deferred work dispatch handler
    - Direct dispatch executes in current thread
    - Deferred dispatch executes in worker thread

Registration and Dispatch

- Work managed at packet granularity
  - E.g., Link layer to protocol dispatch
  - E.g., Tunnel decapsulation dispatch
- Subsystem registers name and handler
  - netisr_register(NETISR_IP, ..., ip_input, ...)
- Packet source picks protocol + dispatch model
  - netisr_dispatch(NETISR_x, m)
  - netisr_queue(NETISR_x, m)
- Deferred processing executes in swi kthread

Use and Abuse

Service name	Use
NETISR_USB	Scheduling only
NETISR_POLL	Scheduling only
NETISR_POLLMORE	Scheduling only
NETISR_ROUTE	Routing socket input
NETISR_ATALK1	Appletalk level 1 input
NETISR_ATALK2	Appletalk level 2 input
NETISR_AARP	Appletalk address resolution input
NETISR_ATM	ATM input
NETISR_ARP	ARP input
NETISR_IP	IPv4 input
NETISR_NETGRAPH	Scheduling only
NETISR_IPV6	IPv6 input
NETISR_IPX	IPX input
NETISR_NATM	NATM input

netisr future

- Direct dispatch now the default in 7-CURRENT
  - Avoid context switches and lowers latency
  - Enables parallelism between network layer input processing across different interfaces, but ...
  - ... disallows parallelism between interface ithread and network layer input processing.
- Netisr2 prototype in Perforce
  - Moves from swis to per-cpu kthreads
  - Will be used for loopback and tunneled traffic
  - Ordering is a key design concern

Key code paths to inspect

- netisr.h
- ip_input.c: ip_init(), ip_input()
- if_ethersubr.c: ether_demux()
- netisr.c:
  - netisr_register(), netisr_unregister()
  - netisr_dispatch(), netisr_queue()
  - start_netisr(), swi_net(), netisr_processqueue()