The ARP state machine

```c
#define NUD_INCOMPLETE 0x01
#define NUD_REACHABLE 0x02
#define NUD_STALE 0x04
#define NUD_DELAY 0x08
#define NUD_PROBE 0x10
#define NUD_FAILED 0x20

/* Dummy states */
#define NUD_NOARP 0x40
#define NUD_PERMANENT 0x80
#define NUD_NONE 0x00

#define NUD_IN_TIMER (NUD_INCOMPLETE|NUD_DELAY|NUD_PROBE)
#define NUD_VALID (NUD_PERMANENT|NUD_NOARP|NUD_REACHABLE|NUD_PROBE|NUD_STALE|NUD_DELAY)
#define NUD_CONNECTED (NUD_PERMANENT|NUD_NOARP|NUD_REACHABLE)
```

Diagram:

- **None**
  - Transition to **Incomplete**
- **Incomplete**
  - Transition to **Reachable**
- **Delay**
  - Transition to **Probe**
  - Transition back to **_Delay**
- **Stale**
  - Transition to **Reachable**
- **Permanent**
  - Transition to **NoArp**
- **NoArp**
  - Transition to **Reachable**
- **Reachable**
  - Transition to **Incomplete**
  - Transition to **NoArp**
- **Probe**
  - Transition to **Reachable**
  - Transition back to **Probe**
When a new route cache element is created, the `arp_bind_neighbour()` function is called. If a suitable ARP cache element cannot be found, `neigh_alloc()` creates a new one and sets its state to NUD_NONE.

When the first packet is sent that requires the new ARP cache element, `neigh_resolve_output()` will be called. This eventually leads to a call to `neigh_event_send`, which sends the ARP request and makes the transition to NUD_INCOMPLETE.

When an ARP response is received, the `struct neighbour` enters the NUD_REACHABLE state.

When `neigh_periodic_timer()`, finds that the last time the entry was confirmed exceed the `reachable` time parameter.

When `neigh_resolve_output()` is eventually called with a `struct neighbour` in the NUD_STALE state.

When the `neigh_timer_handler()` function associated with the `struct neighbour` is called in the NUD_DELAY state, a transition is made to NUD_PROBE and an ARP request is issued.

1  `__neigh_event_send: 707`
   `n->state == NUD_NONE`
3  `neigh_periodic_timer: 598`
   `neigh_sync:537`
   `now-n->confirmed > n->parms->reachable`
4  `__neigh_event_send:740`
   `n->state == NUD_STALE`
6  `neigh_timer_handler: 650`
   `neigh_sync:543`
   `now-n->confirmed < n->parms->reachable`
   `arp_rcv:806`
5  `neigh_timer_handler: 657`
   `n->state == NUD_DELAY`

Note: the textbook claims that receipt of a data packet refreshes the REACHABLE state. 20 extra credit quiz points to the first person who can demonstrate where/how this is done.