Current status of nf_conntrack and some IPv6 improvements

nf_conntrack

Current status

- Synchronizing nf_conntrack with ip_conntrack
  - Thanks a lot for helps on this!
- Solving compatibility issues
  - Common definitions in {ip,nf}_conntrack are unified.
    - ip_conntrack_{info, status, events, stat....}
  - Inline functions to access inner members in conntrack structure are introduced.
  - Users select ip_conntrack or nf_conntrack at compilation.

Remaining issues

- Tuple size – union of IPv4/IPv6 address
  - I believe it's better to keep current structure for simplicity.
  - Or it's better to introduce sockaddr style structure ?

TODO

- nfctnetlink
- IPv4 NAT

Current works on IPv6 improvements

Reducing the number of parsing extension headers

- Issue: Multiple times to parse headers
  - to find transport protocol header. e.g “-p tcp ..."
  - to find extension header which match modules inspect. e.g. “-m ah ..."
- Suggestion
  - Caching the offset to headers while walking table.
    - adding the argument “hints” to match()

Implementing match to compare with final destination address

- ip6tables option "-d" represents
  - Destination address in IPv6 header.
- This may be different with the final destination which receiver process binds.
  - Routing header type 0
  - Routing header type 2 for routing optimization of Mobile IPv6
- Which address user expect ?
  - >The implementing match "finaldst" will help them.

Fin.