LACP (aka bonding)

Link Aggregation, also known as *bonding* in Linux, uses two or more network interfaces to make one *aggregate* interface. This is designed for speed and redundancy.

With Link Aggregation, traffic may be spread across the multiple lines, increasing speed, or increasing fault tolerance by moving data seamlessly from one line to another in case of network failure. Combinations of these are available.

Implementation

An example of setting up an LACP bonded pair in Debian Wheezy can be found in the article Xen Networking

LACP modes

LACP has several different modes which are combinations of the above. Some of the modes require network switches that “talk” LACP, though modes 0, 1, 5 & 6 do not.

balance-rr (mode 0)

```
    Round-robin policy: Transmit packets in sequential
    order from the first available slave through the
    last. This mode provides load balancing and fault
    tolerance.
```

active-backup (mode 1)

```
    Active-backup policy: Only one slave in the bond is
    active.  A different slave becomes active if, and only
    if, the active slave fails. The bond's MAC address is
    externally visible on only one port (network adapter)
    to avoid confusing the switch.
```

```
    This mode provides fault tolerance. The "primary"
    option affects the behavior of this mode.
```

balance-xor (mode 2)

```
    XOR policy: Transmit based on the selected transmit
    hash policy.  The default policy is a simple [(source
```

```
    MAC address XOR'd with destination MAC address) modulo
    slave count].  Alternate transmit policies may be
    selected via the xmit_hash_policy option.
```

```
    This mode provides load balancing and fault tolerance.
```

broadcast (mode 3)

```
    Broadcast policy: transmits everything on all slave
    interfaces.  This mode provides fault tolerance.
```

802.3ad (mode 4)

```
    IEEE 802.3ad Dynamic link aggregation.  Creates
    aggregation groups that share the same speed and
    duplex settings.  Utilizes all slaves in the active
    aggregator according to the 802.3ad specification.
```

```
    Slave selection for outgoing traffic is done according
    to the transmit hash policy, which may be changed from
    the default simple XOR policy via the xmit_hash_policy
    option. Note that not all transmit policies may be 802.3ad
    compliant, particularly inregards to the packet mis-ordering
    requirements of section 43.2.4 of the 802.3ad standard.
    Differing peer implementations will have varying tolerances for
    noncompliance.
```

```
    Note: Most switches will require some type of configuration
    to enable 802.3ad mode.
```

balance-tlb (mode 5)

```
    Adaptive transmit load balancing: channel bonding that
    does not require any special switch support.  The
    outgoing traffic is distributed according to the
    current load (computed relative to the speed) on each
    slave.  Incoming traffic is received by the current
    slave.  If the receiving slave fails, another slave
    takes over the MAC address of the failed receiving
    slave.
```

balance-alb (mode 6)

```
    Adaptive load balancing: includes balance-tlb plus
    receive load balancing (rlb) for IPV4 traffic, and
```

```
    does not require any special switch support.

    When a link is reconnected or a new slave joins the
    bond the receive traffic is redistributed among all
    active slaves in the bond by initiating ARP Replies
    with the selected MAC address to each of the
    clients. The updelay parameter must
    be set to a value equal or greater than the switch's
    forwarding delay so that the ARP Replies sent to the
    peers will not be blocked by the switch.
```

Resources

- http://serverfault.com/questions/446911/what-are-the-differences-between-channel-bonding-modes-in-linux