1toAll Broadcast Hypercube

```
one2allbcast(int n, int rank, void *data, int count,
        MPI_Datatype dtype)
{
    MPI_Status status;
    int mask, partner;
    int mask2 = ((1 << n) - 1) ^ (1 << n-1);
    for (mask = (1 << n-1); mask; mask >>= 1, mask2 >>= 1)
    {
        if (rank & mask2 == 0)
        {
            partner = rank ^ mask;
            if (rank & mask)
                MPI_Recv(data, count, dtype, partner, 99,
                        MPI_COMM_WORLD, &status);
            else
                MPI_Send(data, count, dtype, partner, 99,
                        MPI_COMM_WORLD);
        }
    }
}
```

Generalized 1toAll Broadcast Hypercube

```
one2allbcast(int n, int rank, void *data, int count,
        MPI_Datatype dtype, int source)
{
    vrank =  rank ^ source;
    MPI_Status status;
    int mask, partner;
    int mask2 = ((1 << n) - 1) ^ (1 << n-1);
    for (mask = (1 << n-1); mask; mask >>= 1, mask2 >>= 1)
    {
        if (vrank & mask2 == 0)
        {
            partner = vrank ^ mask;
            if (vrank & mask)
                MPI_Recv(data, count, dtype, partner, 99,
                        MPI_COMM_WORLD, &status);
            else
                MPI_Send(data, count, dtype, partner, 99,
                        MPI_COMM_WORLD);
        }
    }
}
```

Accumulate Hypercube

```
accumulate(int n, int rank, void *data, int count,
        MPI_Datatype dtype, int source)
{
    vrank =  rank ^ source;
    MPI_Status status;
    int mask, partner, mask2 = 0;
    accum(data, count, dtype);
    for (mask = 1; mask; mask <<= 1, mask2 = (mask2 << 1) | 1)
    {
        if (vrank & mask2 == 0)
        {
            partner = vrank ^ mask;
            if (vrank & mask)
            {
                MPI_Recv(data, count, dtype, partner, 99,
                        MPI_COMM_WORLD, &status);
                accum(data, count, dtype);
            }
            else
                MPI_Send(data, count, dtype, partner, 99,
                        MPI_COMM_WORLD);
        }
    }
}
```