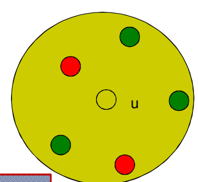
DP and its variations

Today's Menu

- Dominant Pruning (DP)
- Partial Dominant Pruning (PDP)
- Total Dominant Pruning (TDP)
- Multi-cover Dominant Pruning (MDP)

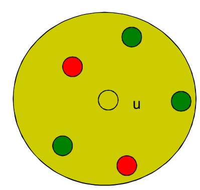
Dominant pruning (DP)

- Proactive
- A node selects some of its neighbors for forwarding to make sure that all of its 2-hop neighbors receive the packet.
- The selected set of nodes are called forward list and specified in the packet header
- While selecting 1-hop neighbors, a node tries to engage as minimum number as possible

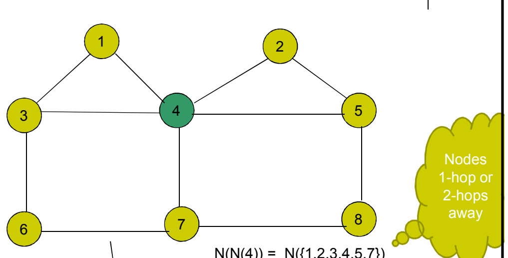


Some terminology

- Needs 2-hop information
- Periodic beacon messages can be used to generate 2-hop neighbor info.
- Let:
  - N(u) = nodes exactly 1-hop away
  - N(N(u))= nodes either 1-hop or 2-hop away
  - N(N(u)) – N(u) = nodes exactly 2-hops away



Example



N(4) = {1,2,3,4,5,7}

N(N(4)) = N({1,2,3,4,5,7})

= {exactly 1-hop} U {exactly 2-hop}

= {1,2,3,4,5,7} U {6,8}

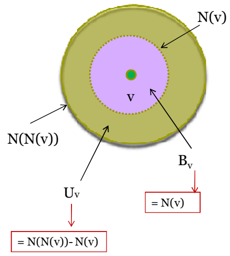
= {1,2,3,4,5,6,7,8}

Dominant pruning explained

- Two cases
  - The originator
  - Intermediate forwarding node

Originator case explained

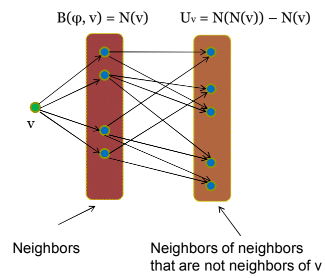
- Suppose node *v* is the originator of a broadcast packet



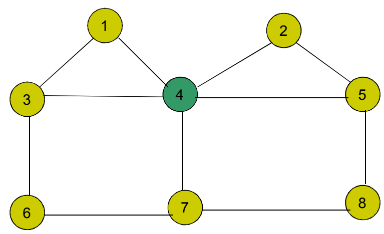
- Node *v* creates a list of neighbors who should forward after receiving from v.

- Here is how *v* creates the forward list
  - All its 1-hop neighbors in *N(v)* will receive when *v* will broadcast.
  - Among its all 1-hop and 2-hop neighbors the following set of nodes U_v haven't received yet: U_v= N(N(v))- N(v)
  - The forward list is the minimum subset of B_v such that each node in U_v is neighbor of some node in B_v.

Originator case explained



DP Algorithm in action

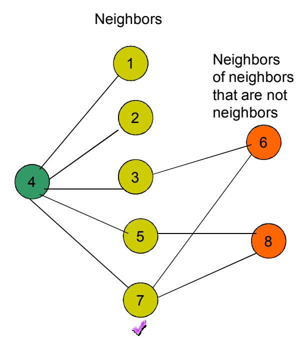


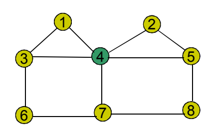
N(4) = {1,2,3,4,5,7}

N(N(4)) = N(4) U {6,8}

={1,2,3,4,5,6,7,8}

DP Algorithm in action



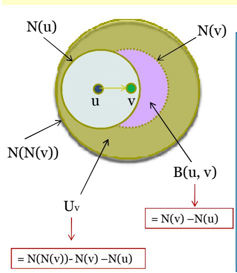


Two solutions: {3,5} and {7}

Engage as minimum as possible

Intermediate Forwarding node case

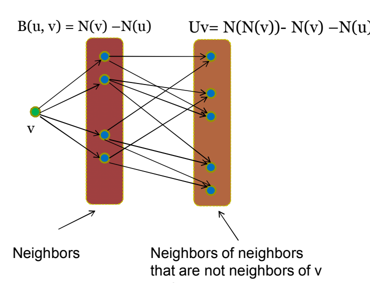
- Suppose node *v* received a broadcast from *u*.



- Node *v* creates a list of neighbors who should re-forward after receiving from v.

- Here is how *v* creates the forward list
  - Among all its 2-hop neighbors, nodes in *N(u)* already received the broadcast from *u*.
  - All its 1-hop neighbors in *N(v)* will receive when *v* will broadcast.
  - Among its all 1-hop and 2-hop neighbors the following set of nodes U_v haven't received yet: U_v= N(N(v))- N(v) –N(u)
  - The forward list is the minimum subset of B(u,v) such that each node in U_v is neighbor of some node in B(u,v).

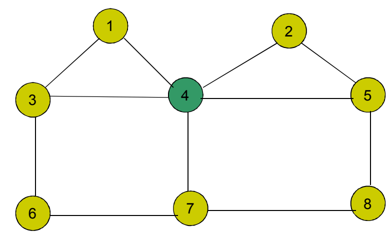
Can be mapped to a set cover problem



DP algorithm

1. Let $F = \emptyset$, $K = \{S_1, S_2, \cdots, S_n\}$ where $S_k = N(v_k) \cap U (1 \le k \le n)$, $Z = \emptyset$.
2. Find the set S_k whose size is maximal in a set K.
3. $F = F \cup \{v_k\}$, $Z = Z \cup S_k$, $K = K - \{S_k\}$, $S_l = S_l - S_k$ for all $S_l \in K$.
4. If $Z = U$, complete the algorithm.
5. Otherwise, repeat from 2 again.

DP Algorithm in action

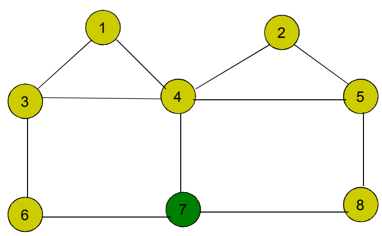


N(u) = {1,2,3,4,5,7}

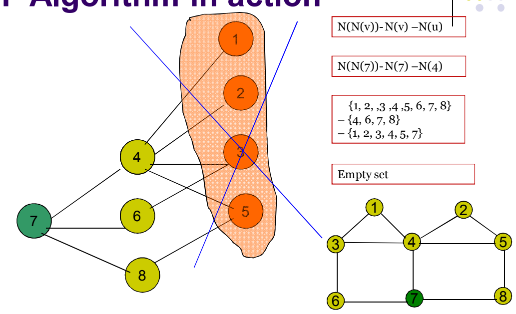
N(N(u)) = N(u) U {6,8}

={1,2,3,4,5,6,7,8}

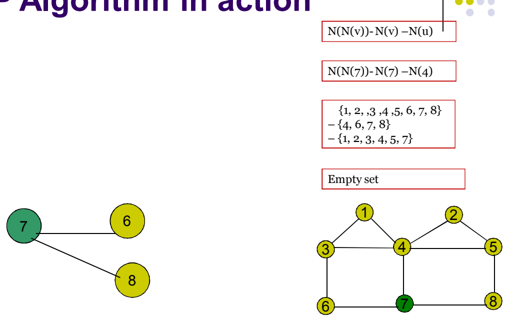
DP Algorithm in action (for Node 7)



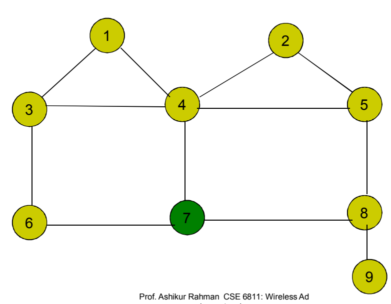
DP Algorithm in action



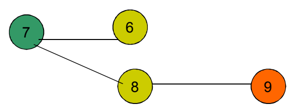
DP Algorithm in action



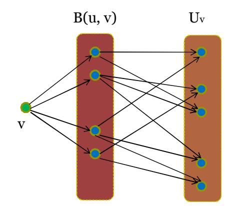
A slight modification



DP Algorithm in action

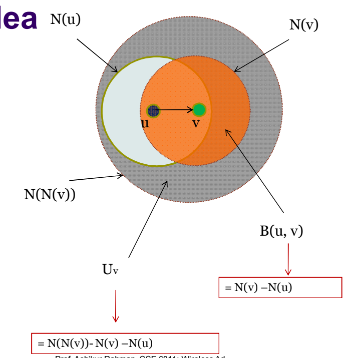


Further optimization in DP!



Optimization is only possible if we can reduce size of U set !

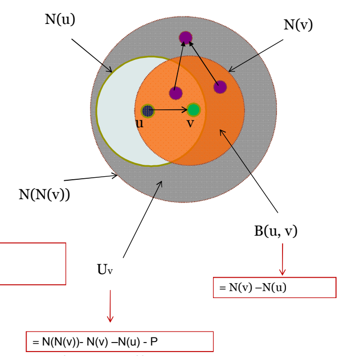
Basic Idea



PDP

Case 1: Neigbor of a common neighbor

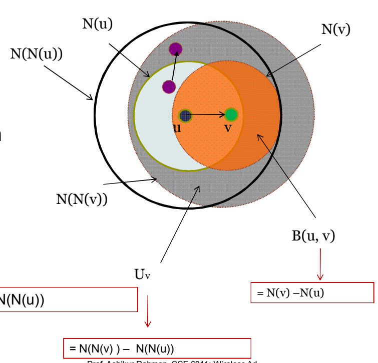
X = N(u) ∩ N(v)
P = N(X)



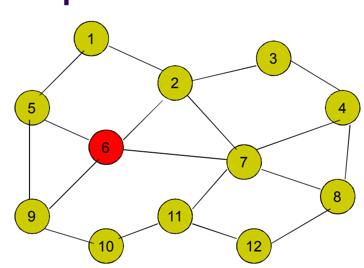
TDP

Case 2: Common in both N(N(u)) and N(N(v))

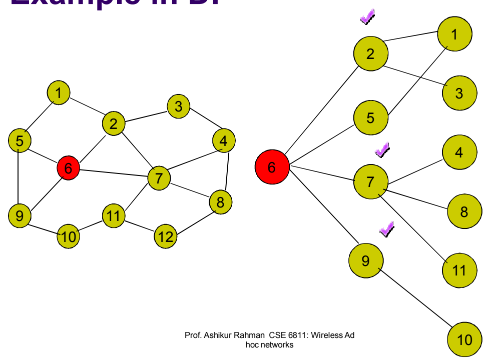
U = N(N(v)) – N(N(u))



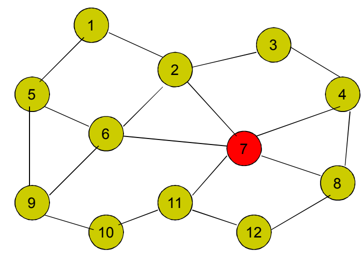
Example in DP



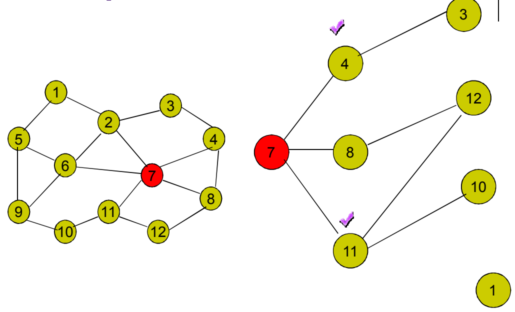
Example in DP



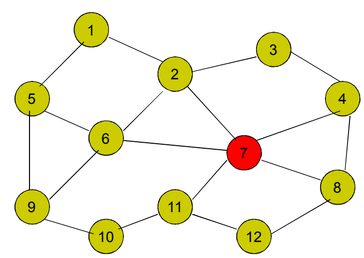
Example in DP



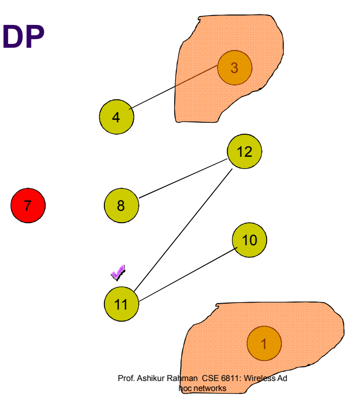
Example in DP



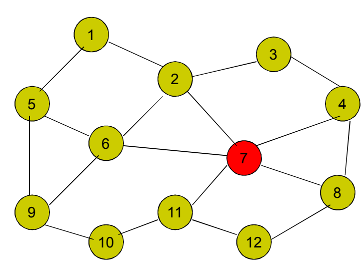
In PDP



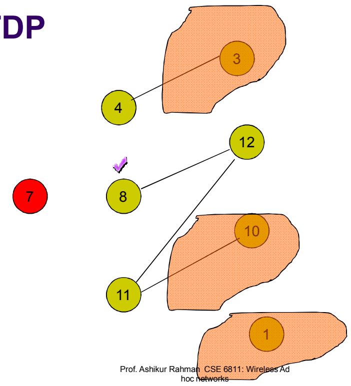
In PDP



In TDP



In TDP



Is redundancy always a bad thing?

Of course not! (think about "un-trusted" environment)

Untrusted Network

- Network with selfish nodes
- Noisy environment
- Network with intentionally un-coopeartive nodes

Example: Noisy environment:

Level of Arsenic monitoring in Bangladesh.

A joint project by UCLA, MIT and BUET.

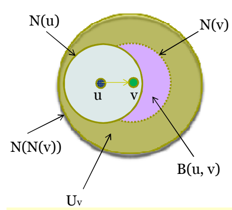
Trade off game

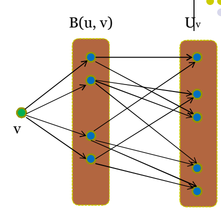
- Un-trusted environment demands fault tolerant design of protocols!
- To make broadcasting algorithms fault tolerant we need *increased* redundancy!

Minimize Redundancy ⟷ Increase Fault-Tolerance

- Wouldn't be nice if we could control redundancy with the help of some tunable parameter(s)?

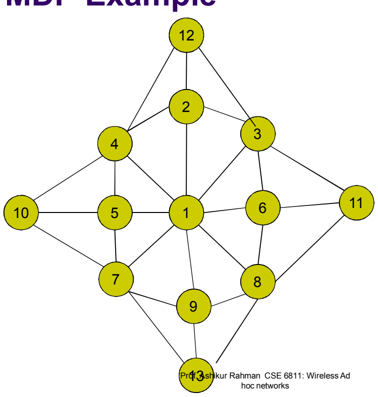
Multi-cover Dominant Pruning (MDP)



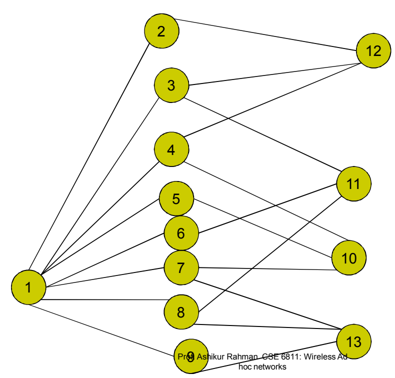


- Reformulate the problem as a set-multicover problem to make it fault-tolerant.
- Now the forward list is the minimum subset of B(u, v) such that each node in U_v is neighbor of at least *m* nodes in B(u,v).
- *m* becomes the tunable parameter. With *m = 1* the algorithm becomes dominant pruning and *m* =∞ the algorithm becomes simple flooding.

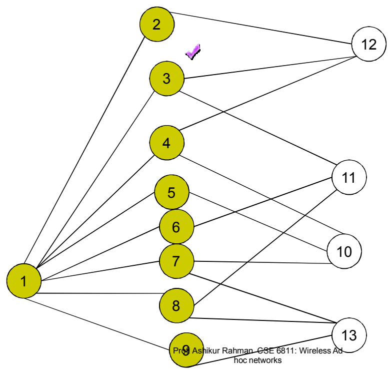
MDP Example



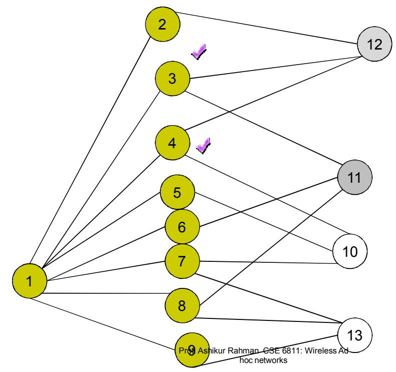
MDP Example



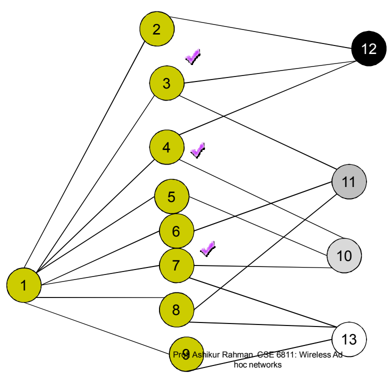
MDP Example



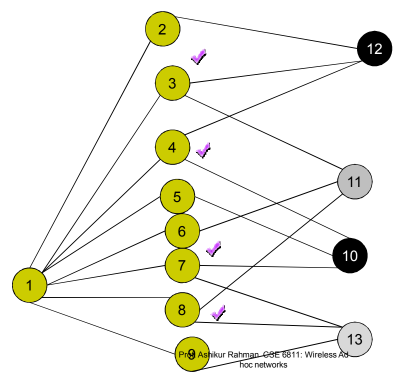
MDP Example



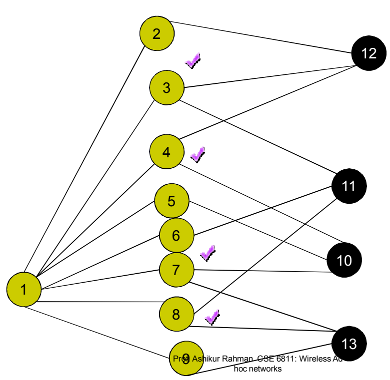
MDP Example



MDP Example



MDP Example



MDP Example