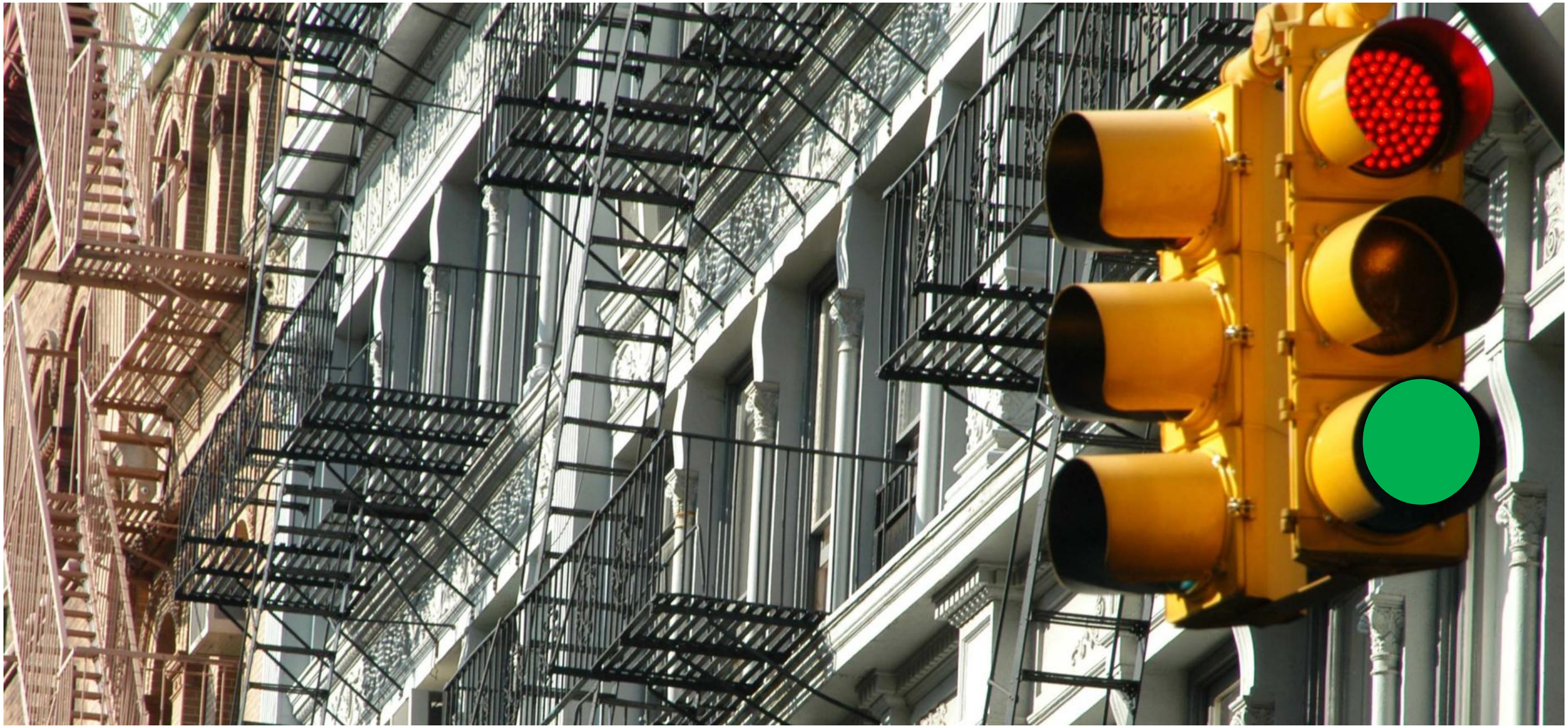


Design of Cross-Domain Overlay Networks or Interdomain Sponsored Content (work in progress)

Gordon Wilfong (Bell Labs)

FND Workshop, July 2013

Joint work with M. Dinitz (Weizmann Inst.) and M. Schapira (Hebrew Univ.)

Alcatel·Lucent

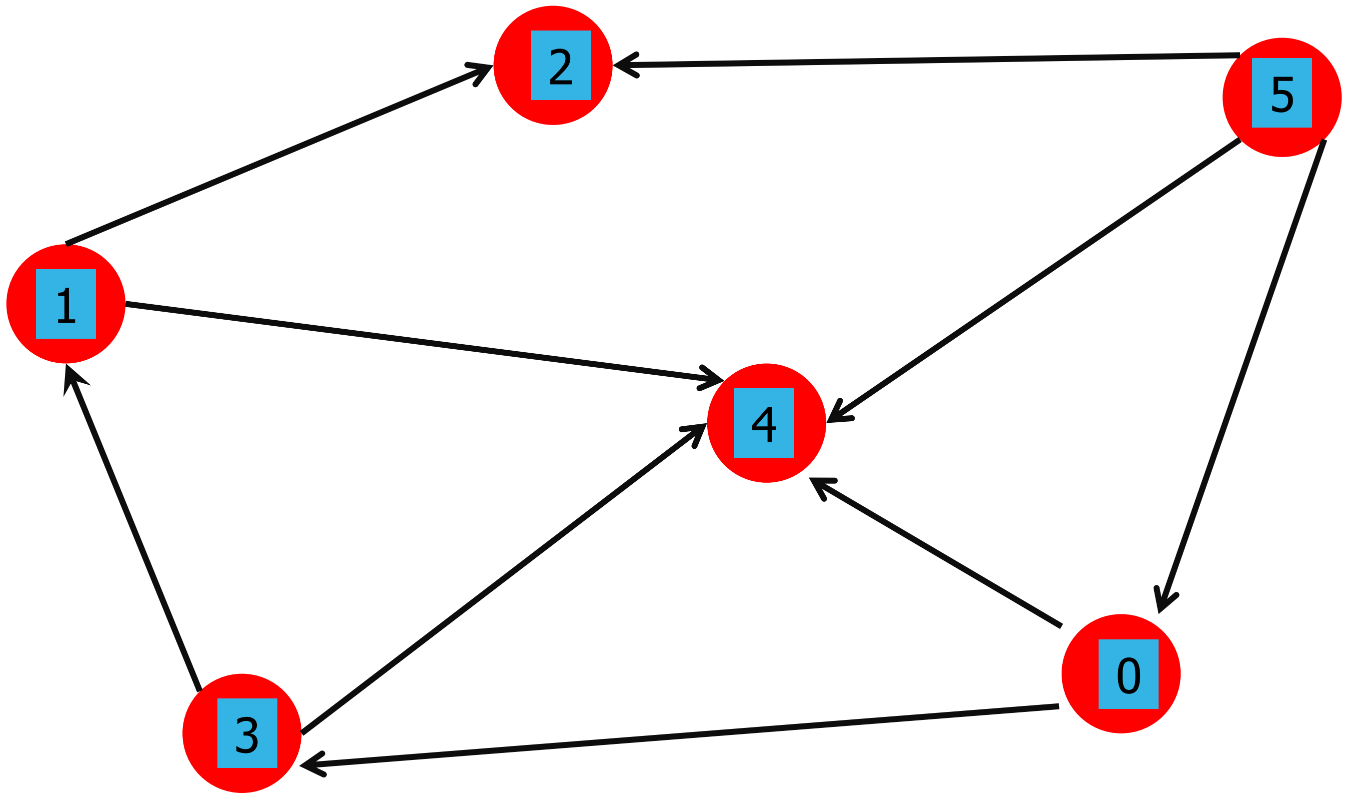
Interdomain Sponsored Content

- 1-800 … (corporations cover customer's long distance charges)
- sponsored content for wireless (content providers cover airtime charges)
- interdomain sponsored content (content providers cover costs of intermediate sub-networks (i.e., domains))

Service Level Agreements (SLAs)

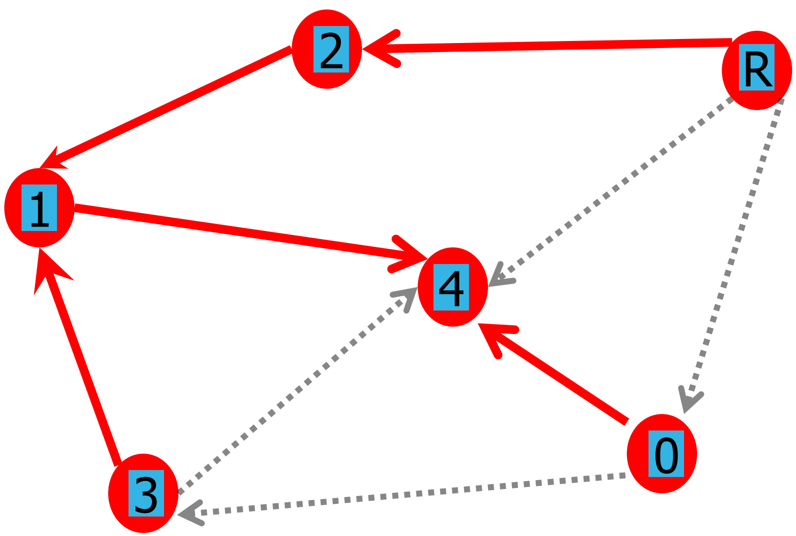
- SLAs are contracts between Autonomous Systems (ASes) to transit traffic
- AS 1 agrees to pay AS 2 for any traffic between 1 and 2
  - AS 1 is the "customer" of "provider" AS 2

Network of SLAs



Subsidized Multi-domain Overlay Network

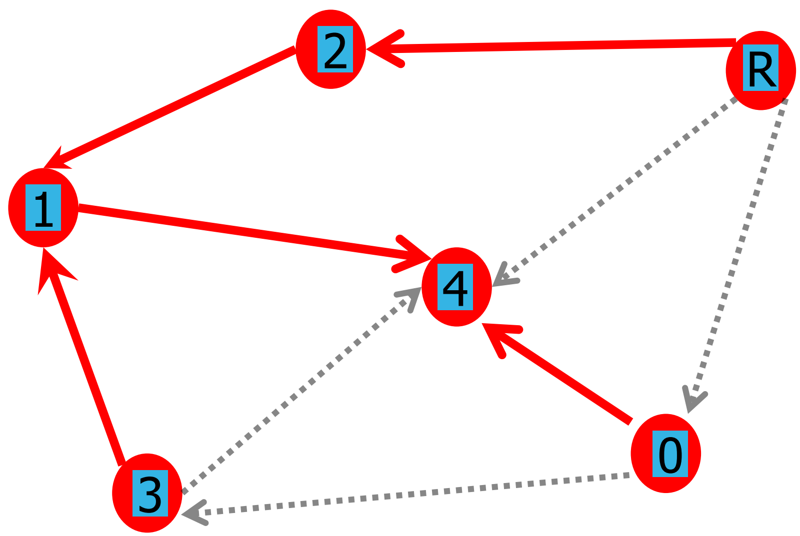
Suppose AS R (e.g., Facebook, Netflix) wants to subsidize other ASes for carrying traffic to/from it to encourage sub-networks to carry content from/to AS R.



Model 1

Root AS R wants to establish subsidized overlay tree.

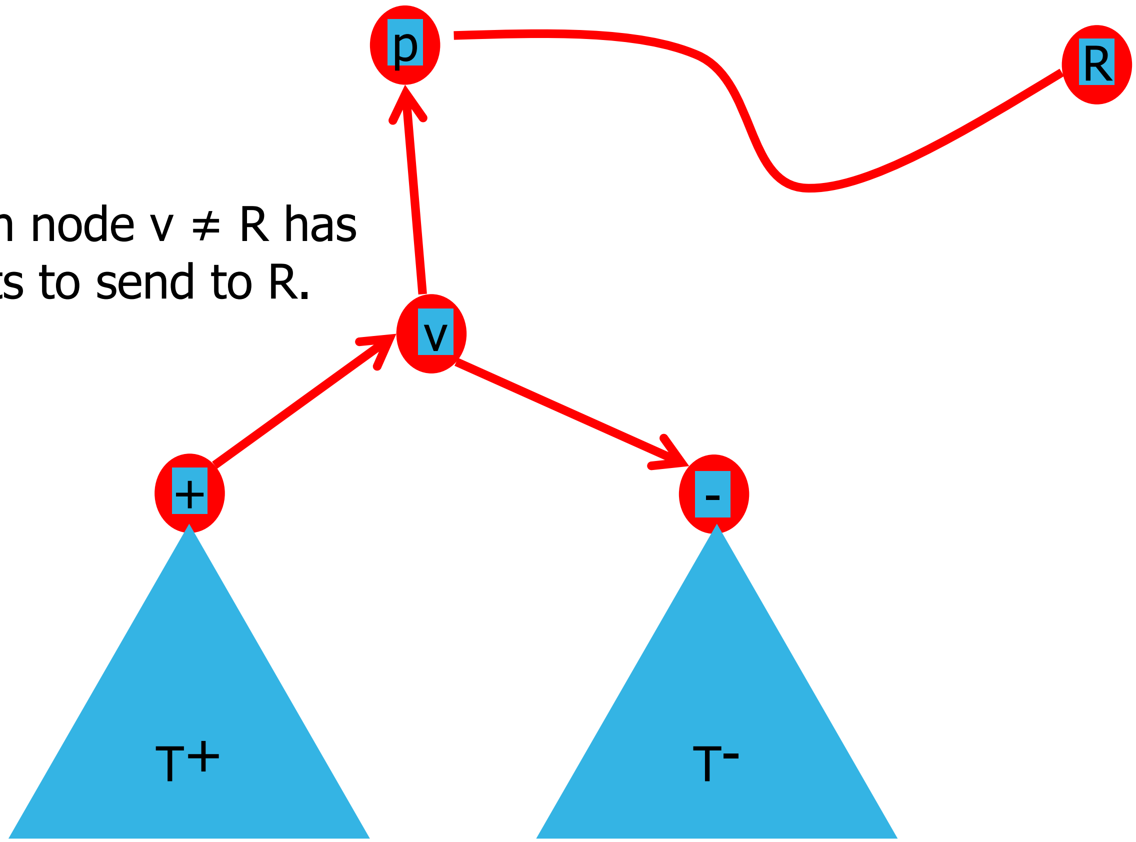
Root R pays other ASes that incur charges for traffic to/from R.



AS 1 gets charged for all traffic between R and ASes 1 and 4 but gets paid for traffic between R and AS 3.

Costs (parent p is a provider)

Assume each node $v \neq R$ has $w(v)$ packets to send to R.

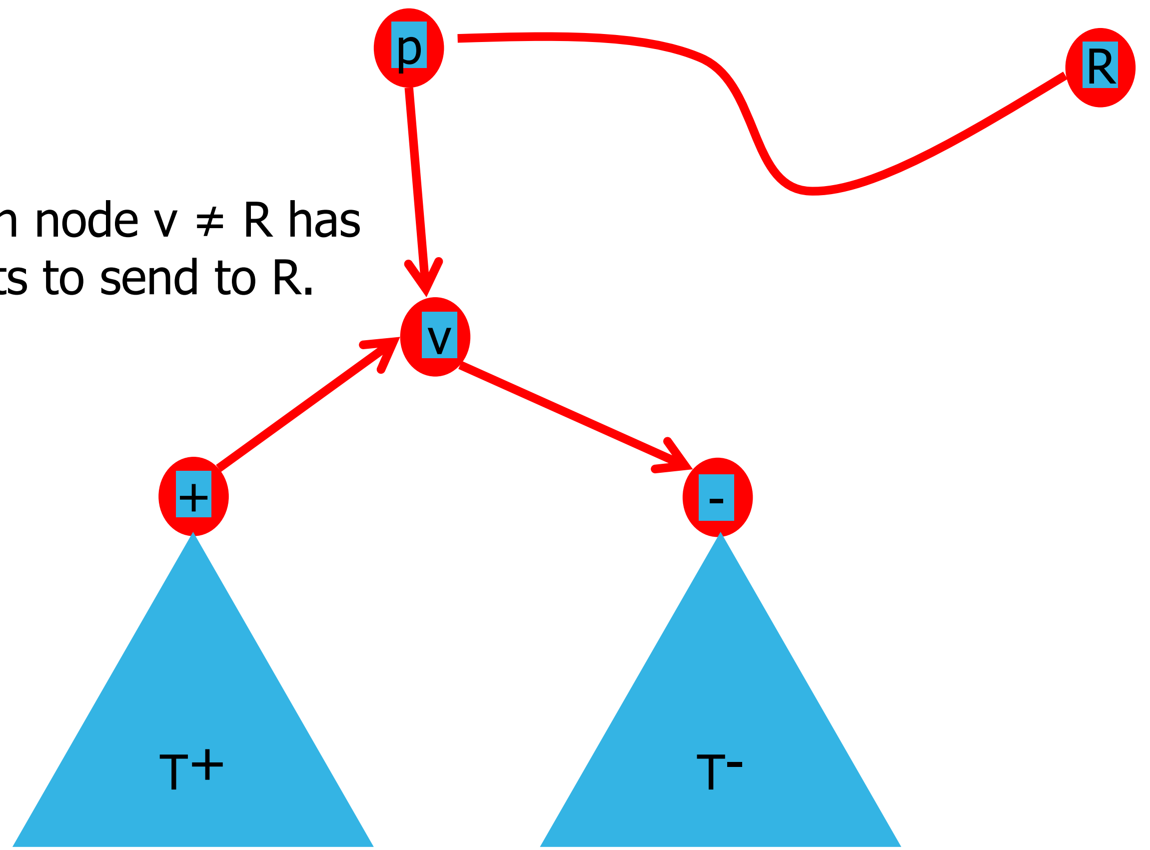


v incurs cost = $w(T^-) - w(T^+) + w(v) + w(T^+) + w(T^-) = 2w(T^-) + w(v)$ [LOSS for v]

$cost(v) = 2w(T^-) + w(v)$

Costs (parent p is a customer)

Assume each node $v \neq R$ has $w(v)$ packets to send to R.



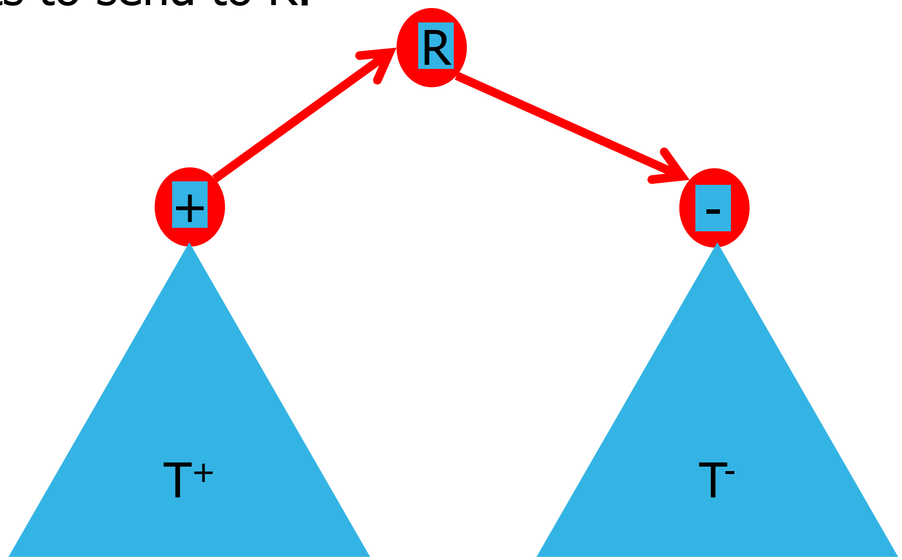
v incurs cost $= w(T^-) - w(T^+) - w(v) - w(T^+) - w(T^-) = -2w(T^+) - w(v) < 0$ [GAIN for v!]

cost(v) = 0

Alcatel·Lucent

Cost of Root R

Assume each node $v \neq R$ has $w(v)$ packets to send to R.



$$\text{R incurs cost} = \begin{cases} w(T^-) - w(T^+), & \text{if } w(T^-) - w(T^+) > 0 \\ 0, & \text{o.w.} \end{cases}$$

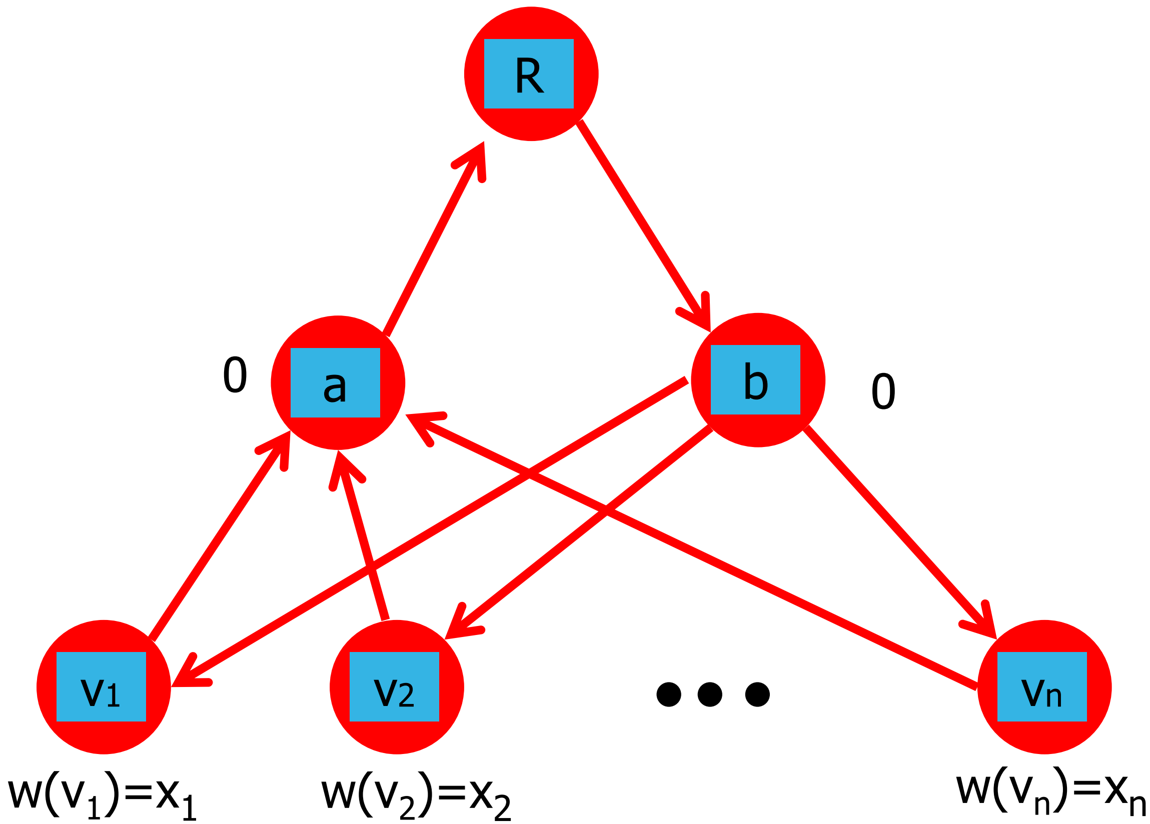
$\mathrm{cost}(R) = \max\{w(T^-) - w(T^+), 0\}$

Cost of Overlay Network T

$$\text{cost}(T) = \sum_{v \in V} \text{cost}(v)$$

Problem: Design a tree network with minimum cost

Weighted Case is NP-complete



$$\text{PARTITION:}\quad \exists ?\ U \subseteq V\ :\sum_{x_i \in U} x_i = \sum_{x_i \in V \setminus U} x_i$$

Uniform Case

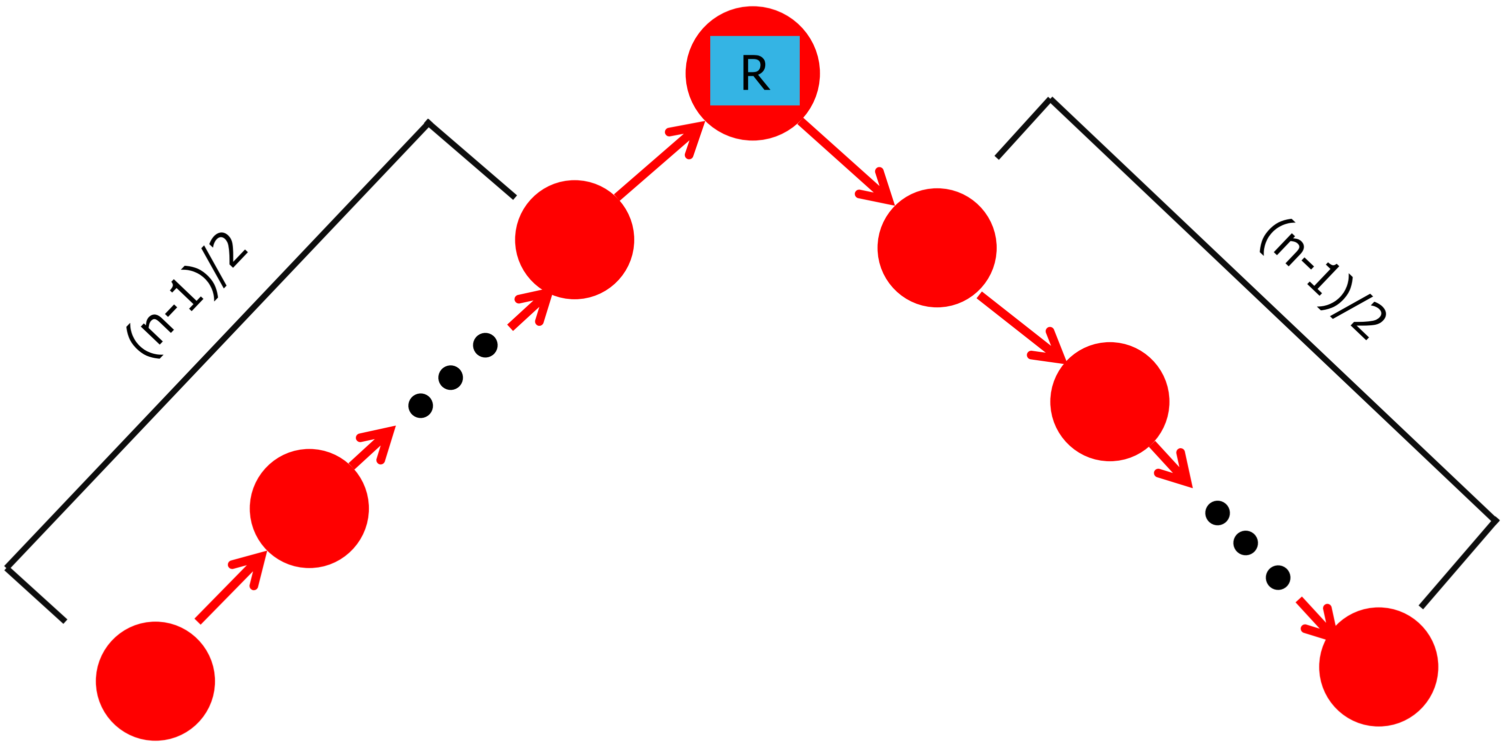
All nodes $v \neq R$ have $w(v)=1$.

Lower Bound

Theorem: $cost(T) \geq (n-1)/2$, $|V|=n$

Lower Bound is Tight

Theorem: cost(T) ≥ (n-1)/2



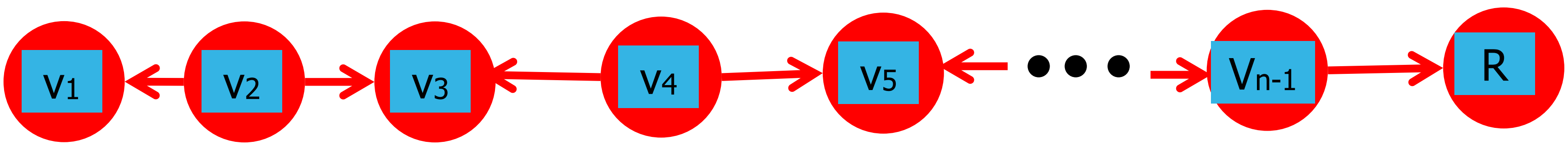
Cost(T) = (n-1)/2

Upper Bound

Theorem: $cost(T) \leq n(n-1)/2$

Upper Bound is Tight

Theorem: cost(T) = n(n-1)/2

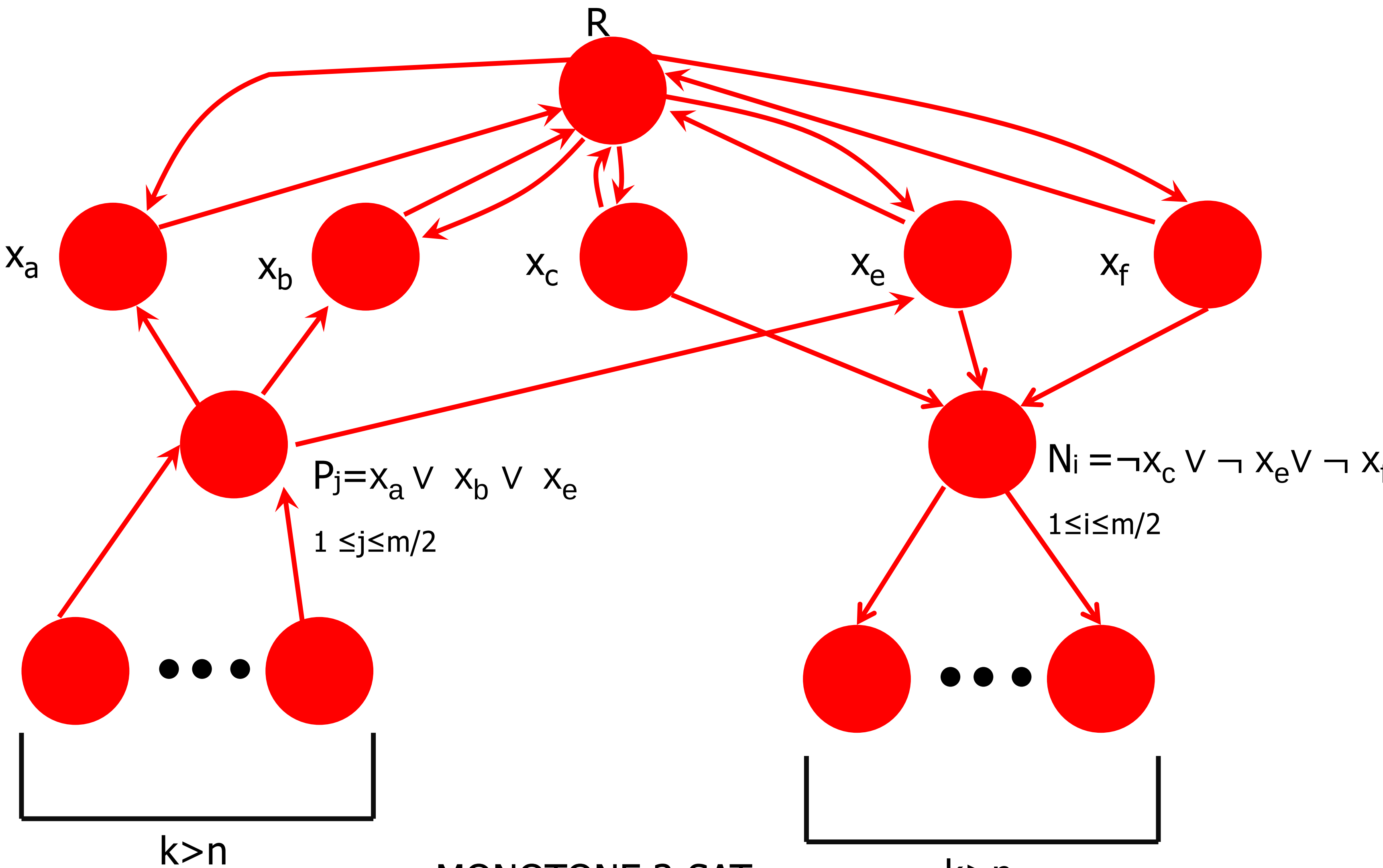


$cost(v_i) = 0$, for odd i

$cost(v_i) = (i-1)+i$, for even i

$$Cost(T) = \sum_{i=1}^{n-1} i = n(n-1)/2$$

Uniform Case is NP-complete

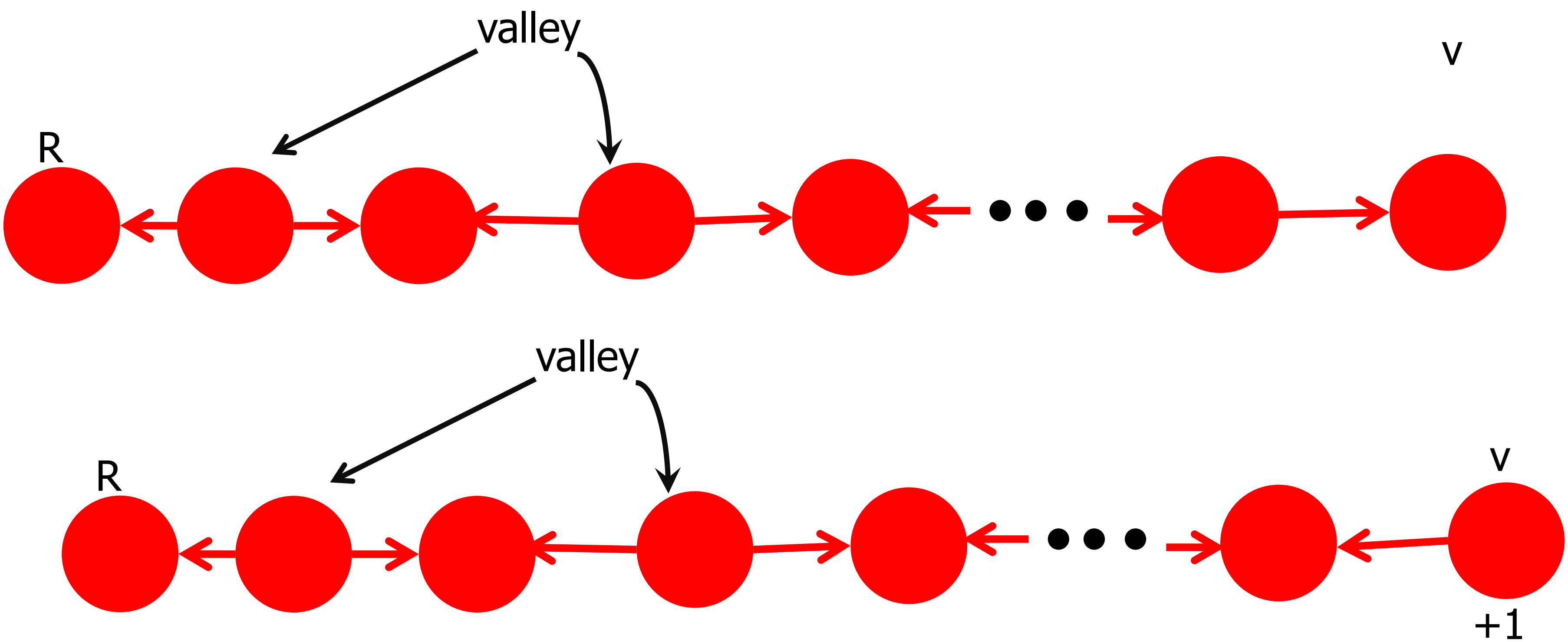


MONOTONE 3-SAT

Path Costs

Define: $C_T(v)$ = 2*(no. valleys) + {1,0}, $v \neq R$

Define: $C_T(R)$ = cost(R)



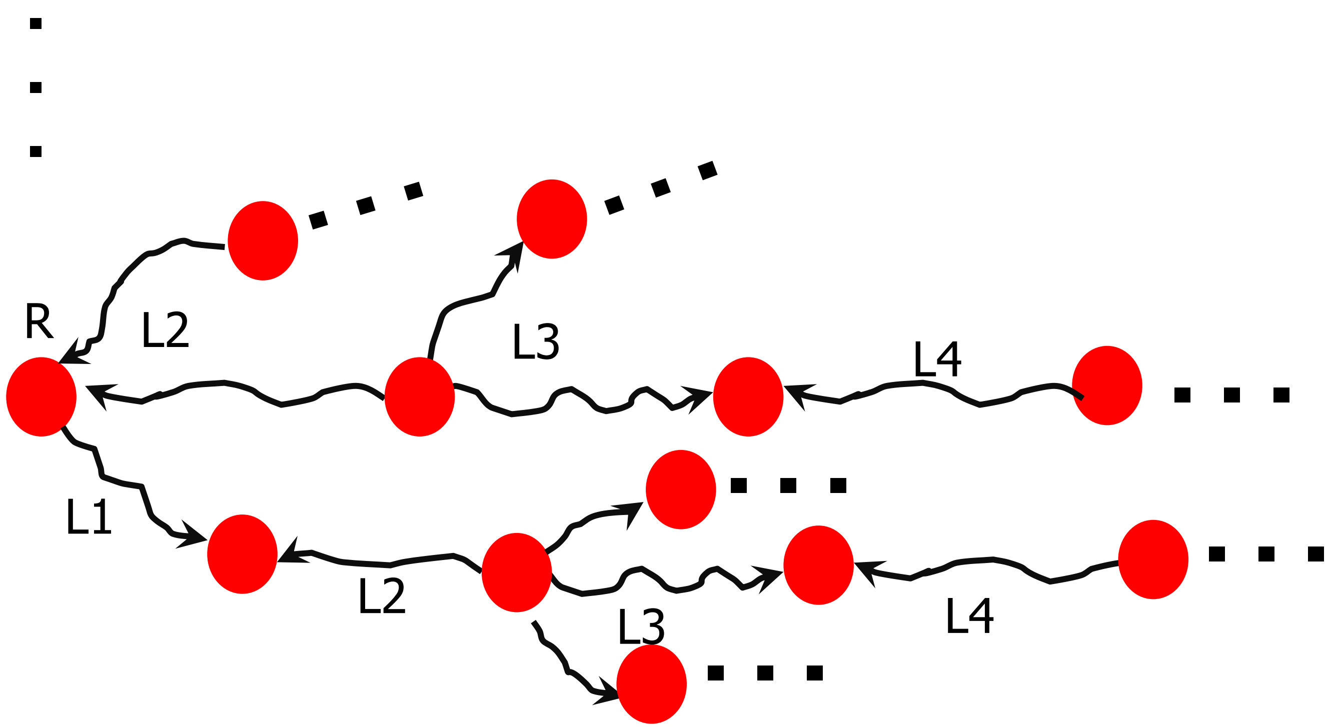
Total Cost

Theorem: $\text{cost}(T) = \sum c_T(v)$

Approximation Algorithm

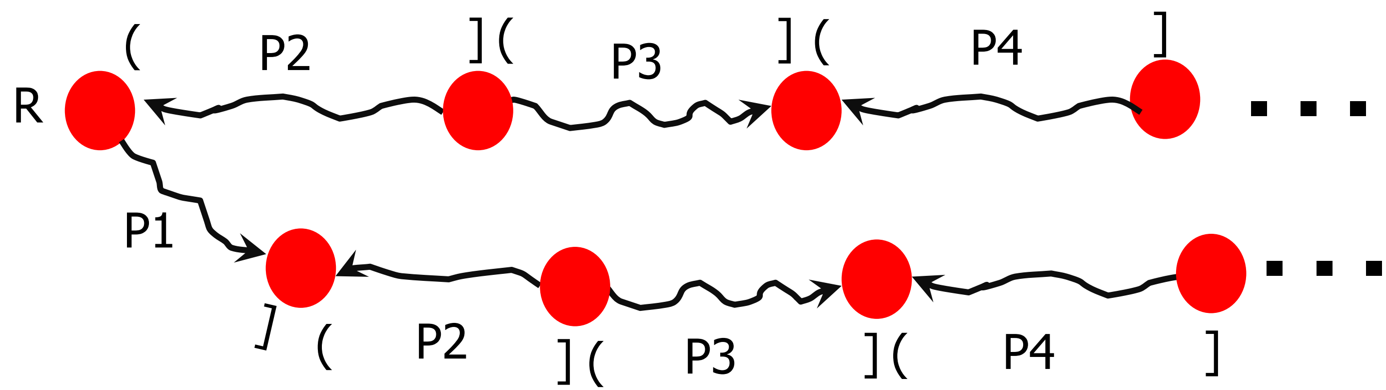
Construct T as follows:
Step 1: Build tree (L1) from R using only customer to provider arcs

Step 2: Build forest (L2) from L1 using only provider to customer arcs



Approximation Algorithm

P = any path from v to R
$P_T(v)$ = path from v to R in T



Theorem: If v ∈ Pi then v ∈ Lj, j ≤ i

Corollary: The number of valleys in $P_T(v)$ is at most the number of valleys in P and $C_T(v) \le C_P(v)$.

Approximation Algorithm

Theorem: T is a 3-approx of optimal tree T*.

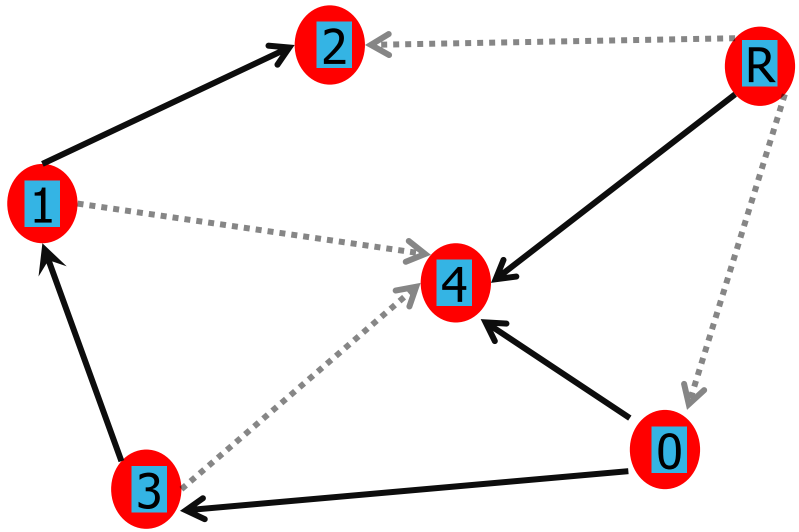
Proof: $C_T(v) \le C_{T*}(v)$

$$cost_T(R) \le n = 2\ n/2 \le 2\ cost(T^*)$$

Model 2

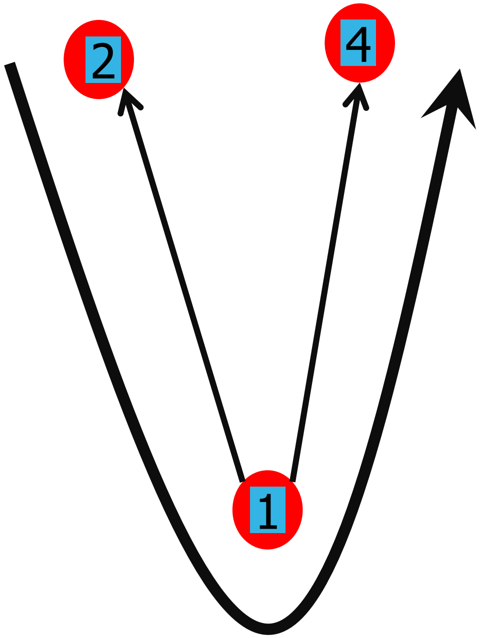
- Assume content provider compensates ASes that incur additional charges AND receives payments from ASes that earn more due to additional traffic.

BGP Routing



- BGP results in a routing tree to each destination
  - e.g., above is a routing tree to AS R.

Valley-free Routing

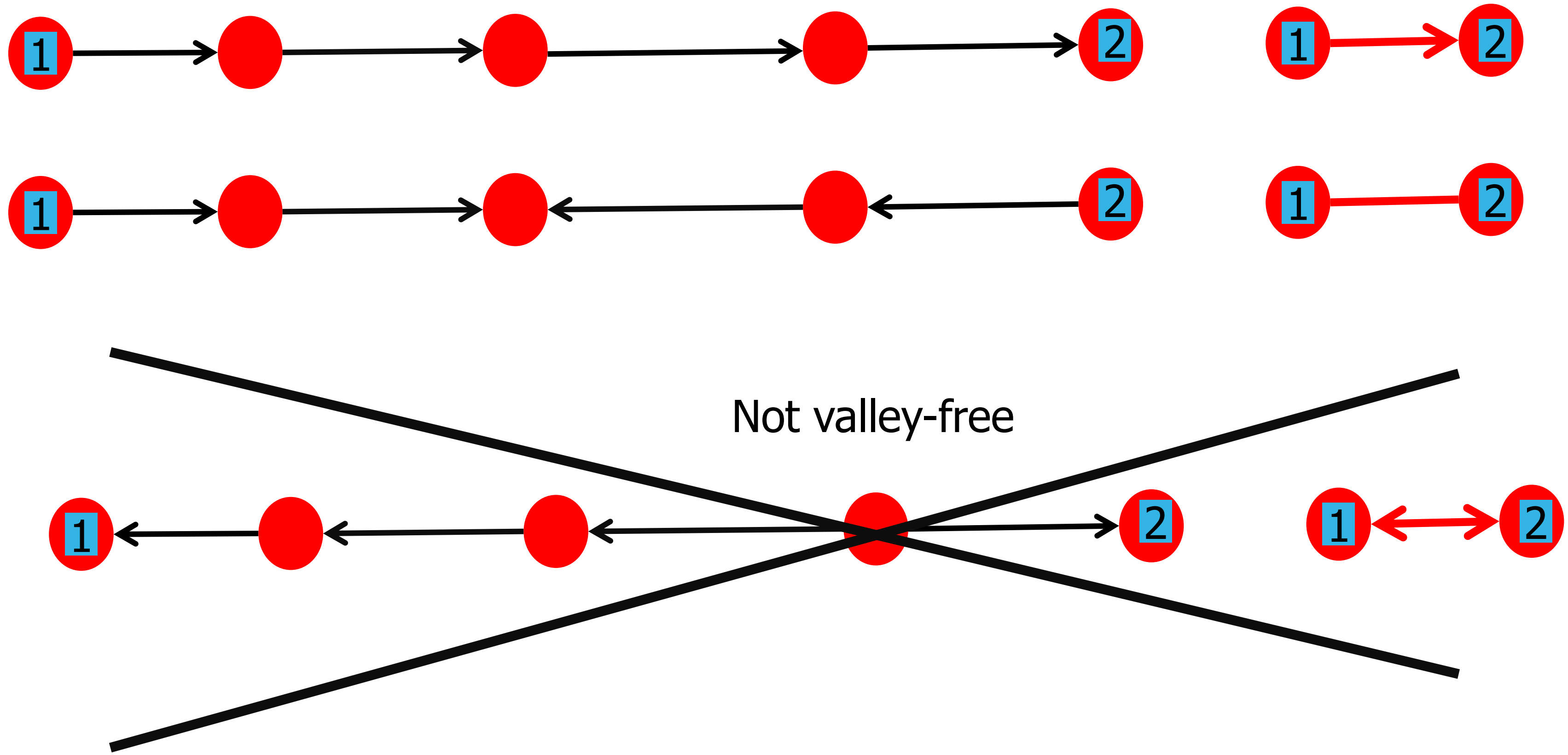


Not valley-free

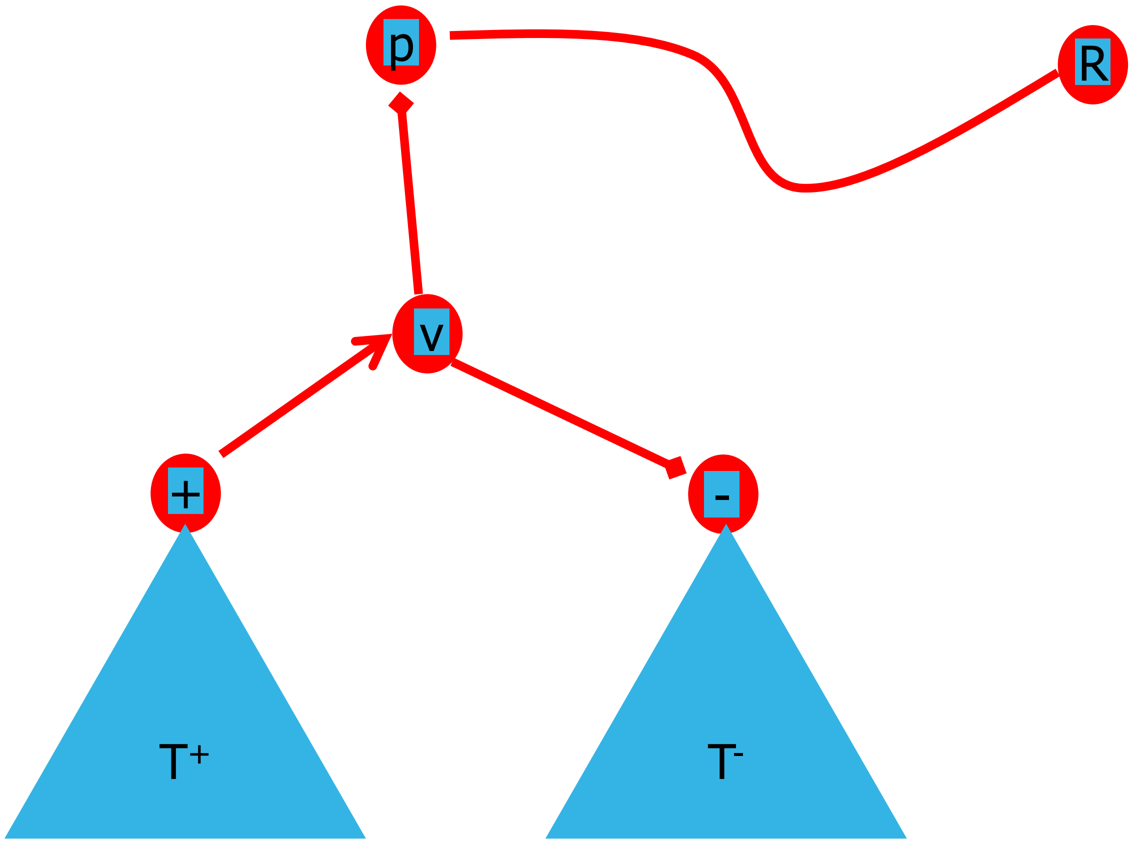
- AS won’t transit traffic for its providers

Edges in Model 2

Root R pays other ASes that incur charges for traffic to/from R over BGP routes.



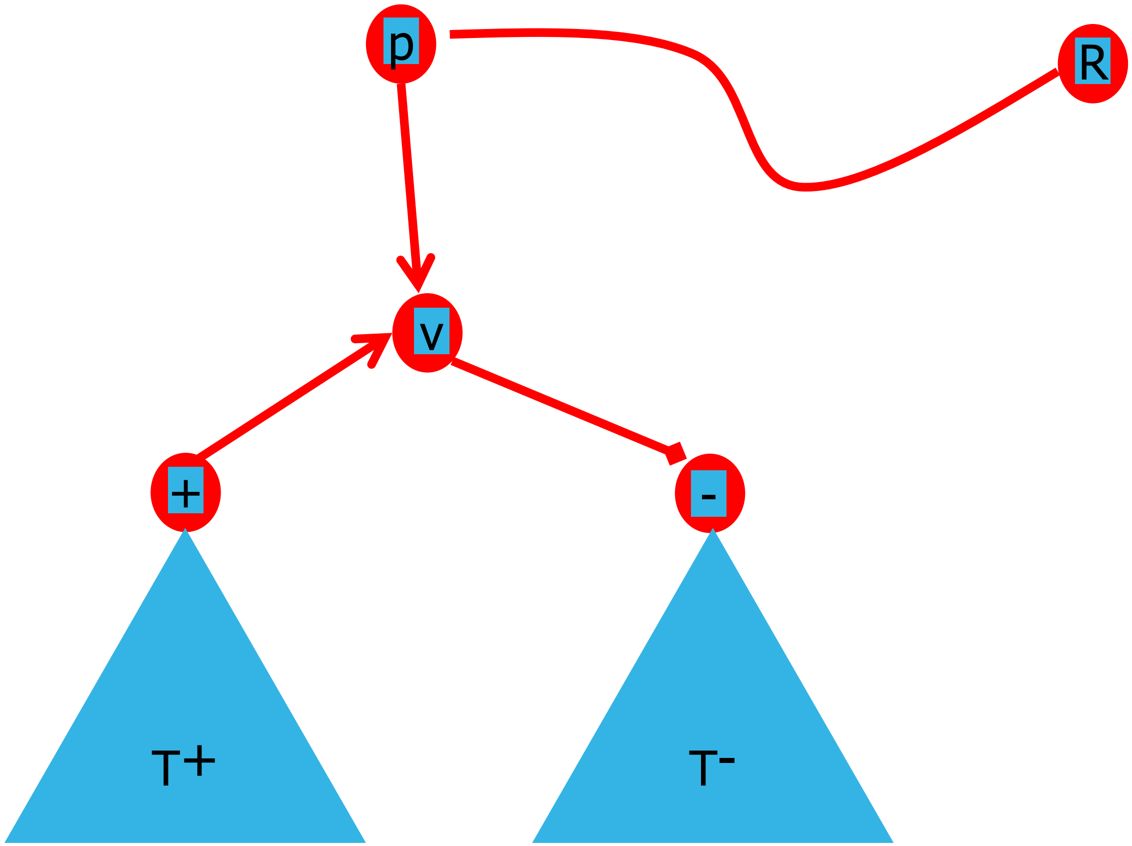
Costs (v is customer)



v incurs cost = $w(T^-) - w(T^+) + w(v) + w(T^+) + w(T^-) = 2w(T^-) + w(v)$ [LOSS for v]

cost(v) = $2w(T^-) + w(v)$

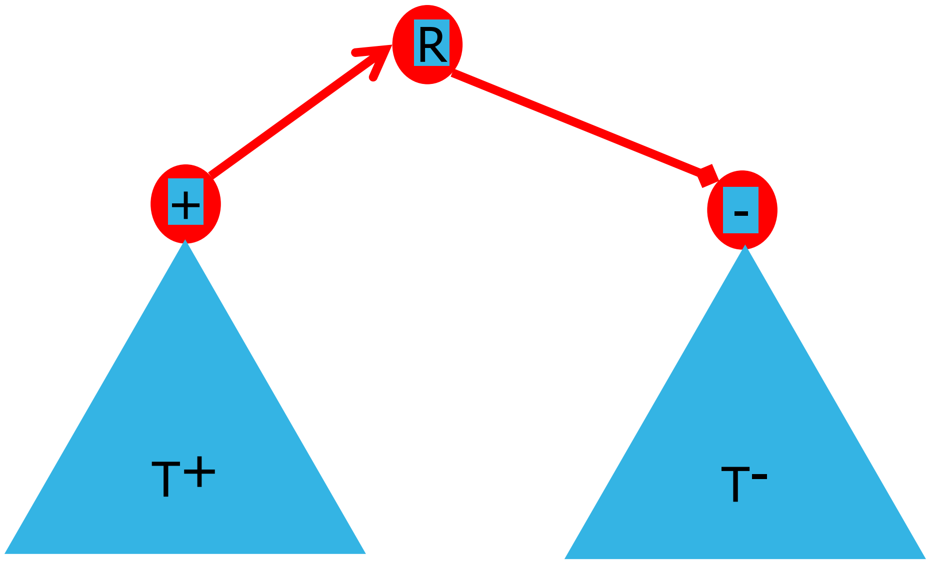
Costs (v is provider)



v incurs cost=w(T^-) – w(T^+) – w(v) – w(T^+) – w(T^-)=-2w(T^+) – w(v)<0 [GAIN for v!]

cost(v) = -2w(T^+) – w(v)

Cost of Root R



R incurs cost = $w(T^-) - w(T^+)$

cost(v) = $w(T^-) - w(T^+)$

Cost of Overlay Network T

$$\text{cost}(T) = \sum_{v \in V} \text{cost}(v)$$

Problem: Design a tree network with minimum cost.

Definitions

- $c_{(v,u)}(u)$ = cost to u when a packet travels from v to u
- $c_{(u,v)}(u)$ = cost to u when a packet travels from u to v
- $l(u,v) = c_{(u,v)}(u) + c_{(u,v)}(v)$ [*length* of arc (u,v)]
- $P_T(u)$ = path in tree T from u to R
- $d(u)$ = length of $P_T(u)$
- $\Gamma_T(u)$ = children of u in T
- $T(v)$ = subtree of T rooted at v
- $p_T(u)$ = parent of u in T

Shortest Path Tree

$$\mathrm{cost}(T) = \sum_{u \in V} [\sum_{v \in \Gamma_T(u)} (c_{(v,u)}(u) \bullet \sum_{x \in T(v)} w(x)) + (c_{(u,p_T(u))}(u) \bullet \sum_{x \in T(u)} w(x))]$$

$$= \sum_{x \in V} w(x) [\sum_{(u,v) \in P_T(x)} (c_{(u,v)}(u) + c_{(u,v)}(v))]$$

$$= \sum_{x \in V} w(x) [\sum_{(u,v) \in P_T(x)} l(u,v)]$$

$$= \sum_{x \in V} w(x) d_T(x)$$

$$p_T(R) = R,$$
$$c_{(R,R)}(R) = 0$$

Note: valley-free paths and economic considerations imply no negative cycles.

Future Work

- Consider case where ASes only pay back a percentage of their gain

- Consider case where packets flow in both directions.

- Use available experimental results deriving relationships between ASes (and pricing data??) to experimentally see how profitable overlays could be.

- Finally, look at bigger problem of cache location.