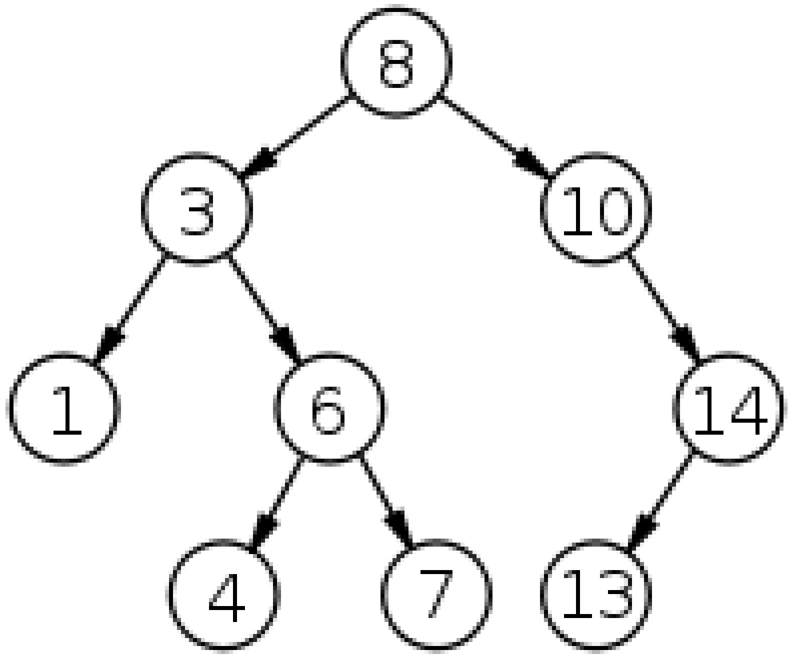
SplayNets

Stefan Schmid, Chen Avin, Christian Scheideler,
Bernhard Häupler, Zvi Lotker

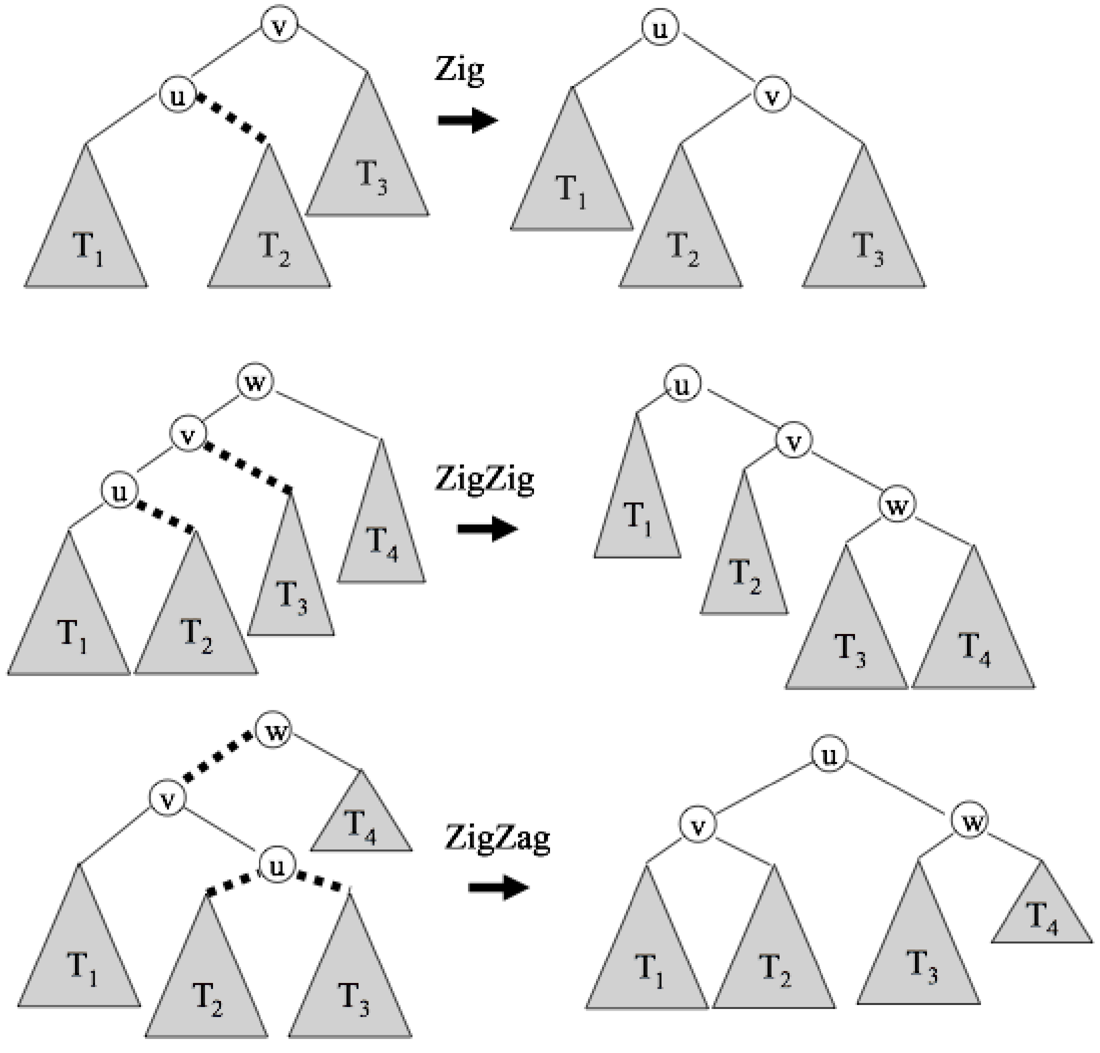
Splay Trees

- Self-adjusting Binary Search Tree [ST85]
- A sequence of lookup requests
- "splay operation" - move requested node to the root
- Frequent nodes get close to the root
- Competitive against optimal static tree



Splay Trees

- Self-adjusting Binary Search Tree [ST85]
- A sequence of lookup requests
- "splay operation" - move requested node to the root
- Frequent nodes get close to the root
- Competitive against optimal static tree



Our Model

- Instead of a set of requests starting at the root, we have a sequence of request pairs

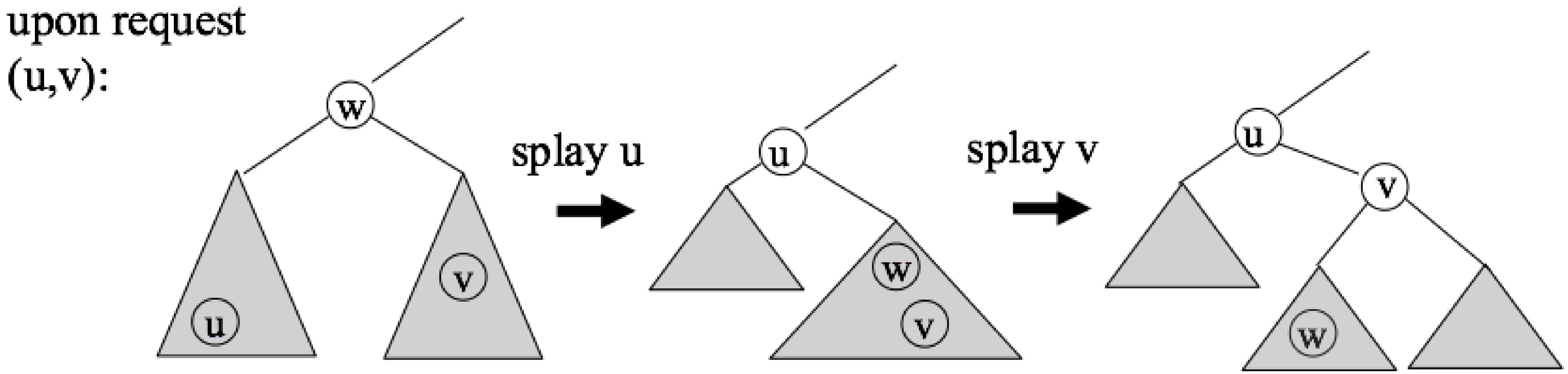
$$\sigma = (\sigma_0, \sigma_1 \ldots \sigma_{m-1}) \quad \sigma_t = (u, v) \in V \times V$$

- What is an optimal static search tree?
- Can be computed in poly time using dynamic programming.
- But that doesn't provide much insight into problem.

Idea: Double Splay

Algorithm 2 Double Splay Algorithm `DS`

(* upon request (u, v) in T *)
$w := \alpha_T(u, v)$ ← Least common ancestor
$T' :=$ **splay** u to root of $T(w)$
splay v to root of $T'(u)$



Analysis: extend Entropy

- Entropy: $X \backsim p(x), \{x_1, \ldots, x_n\}$

$$H(X) = \sum_{i=1}^{n} p(x_i) \log_2 \frac{1}{p(x_i)} \leq \log n$$

- Empirical Entropy: $\hat{X}(\sigma) = \{f(x_1), \ldots, f(x_n)\}$

$$H(\hat{X}), H(\hat{Y}), H(\hat{X}, \hat{Y}), H(\hat{X}|\hat{Y})$$

Our Results

Theorem: Let σ be an arbitrary sequence of communication requests, then for any initial BST T_0,

$$\mathrm{Cost}(DS, T_0, \sigma) = O(H(\hat{X}) + H(\hat{Y}))$$

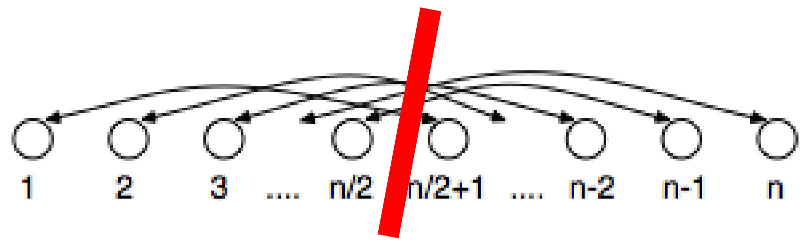
where $H(\hat{X})$ and $H(\hat{Y})$ are the empirical entropies of the sources and the destinations in σ, respectively. Moreover, for any optimal BST network T,

$$\mathrm{Cost}(T, \sigma) = \Omega(H(\hat{Y} \mid \hat{X}) + H(\hat{X} \mid \hat{Y})).$$

Further Results

- A cut based lower bound: there is a cut C

$$\text{Cost}(T,\sigma) = \Omega(H(C(\sigma)))$$



- An expansion lower bound
- Optimality in special cases:
  - Requests form product distribution, rooted binary tree, laminated sets, ...

Thank you!