Online
Path Computation & Function Placement in SDNs

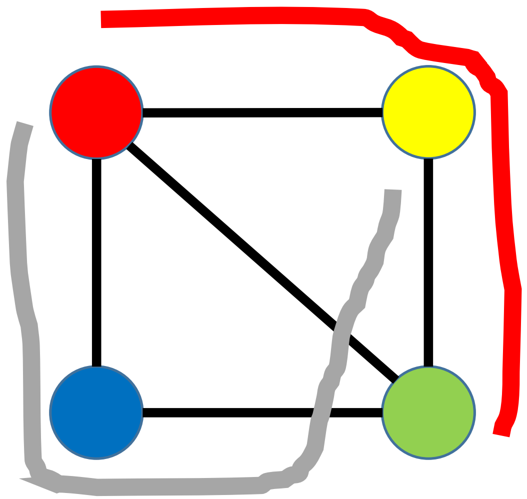
Guy Even
Tel Aviv University

Moti Medina
MPI for Informatics

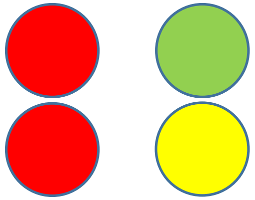
Boaz Patt-Shamir
Tel Aviv University

Today’s Focus: Online Virtual Circuit Routing

- Network: $G = (V, E, c)$
  - V is the set of nodes
  - E is the set of links
  - $c : E \to \mathbb{N}^+$ edge capacities
- Requests: paths $r_j = \langle s_j, t_j, b_j, d_j, [\alpha_j, \beta_j] \rangle$
  - s_j source node
  - t_j destination node
  - b_j benefit (per time step)
  - d_j BW requirement
  - $[\alpha_j, \beta_j]$ arrival/depart times
    - β_j is either known or unknown upfront.



$r_1 = \langle \quad \rangle$
$r_2 = \langle \quad \rangle$



$load(e) \triangleq flow(e)/c(e)$

Performance Measure - Competitive Ratio [Sleator, Tarajan 85]

- ALG : Online alg
- σ : sequence of path requests
- $ALG(\sigma)$: total benefit due to σ
- $OPT(\sigma)$: Max benefit from σ by a feasible allocation

$$\boldsymbol{\rho(ALG) \triangleq \inf_{\sigma} \frac{ALG(\sigma)}{OPT(\sigma)}}$$

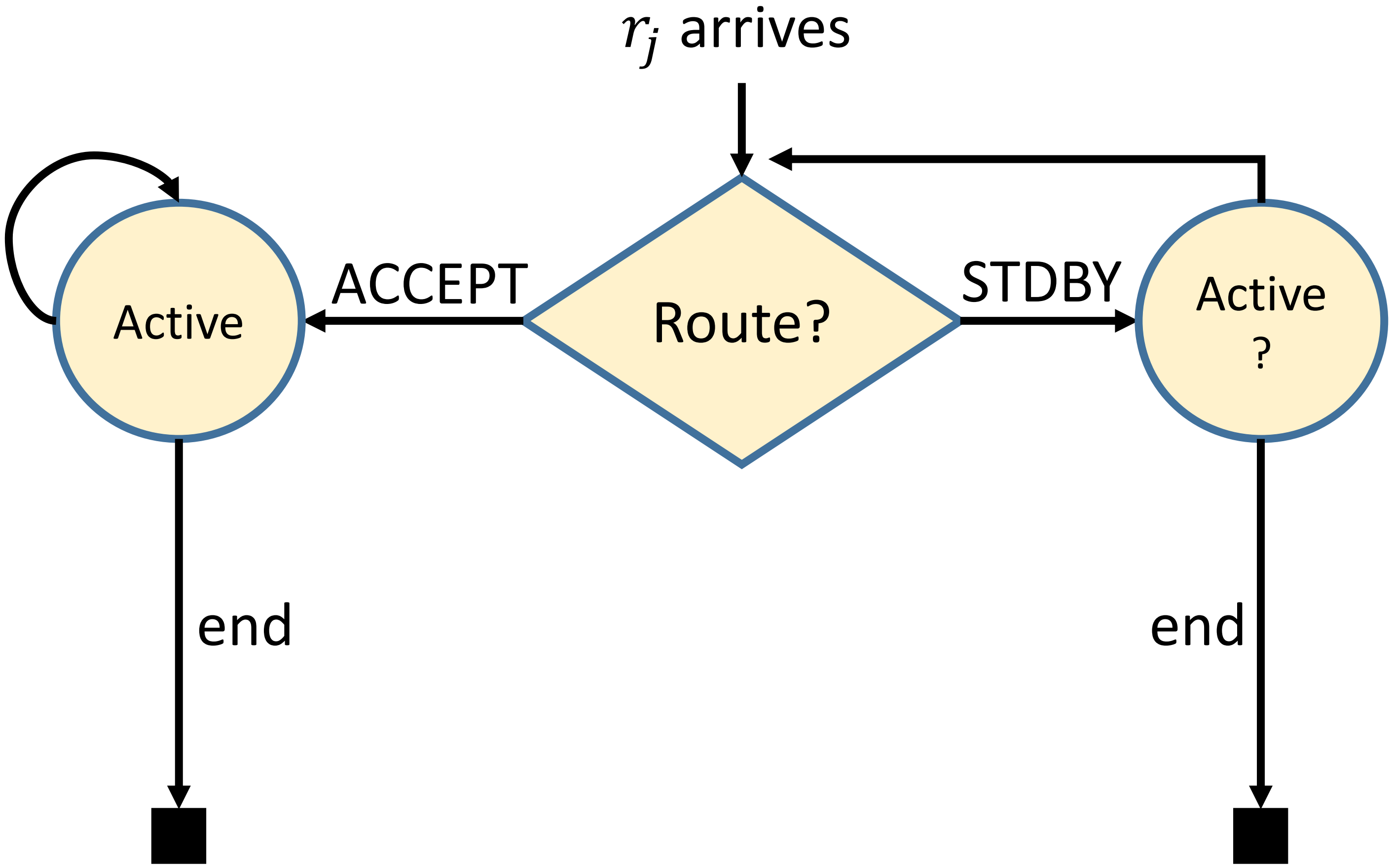
Online Versions

		Model	
		ACC/REJECT	[redacted]
Duration	known	[AAP93] $C.R. = \log(\lvert V\rvert b_{\max})$ Tight	[redacted] ?
	unknown	Unbounded $C.R. \geq time$	[redacted]

Benefits:

- ACC/REJECT $= \sum_{accepted} b_j$
- [redacted] ?
- Known duration $\approx$ only arrivals (persistent requests)

Proposal: ACC/STDBY model



- Unknown duration
- No preemption (!)
- No rerouting (!)
- Request pays b_j per served time unit.

- **Question: what about online competitive analysis?**

Online Versions

		Model	
		ACC/REJECT	ACC/STDBY
Duration	known	[AAP93] $C.R. = \log(\lvert V \rvert b_{\max})$ Tight	Same as ACC/REJECT
	unknown	Unbounded $C.R. \geq time$	**New**: $C.R. = \log(\lvert V \rvert b_{\max})$ Tight

Benefits:

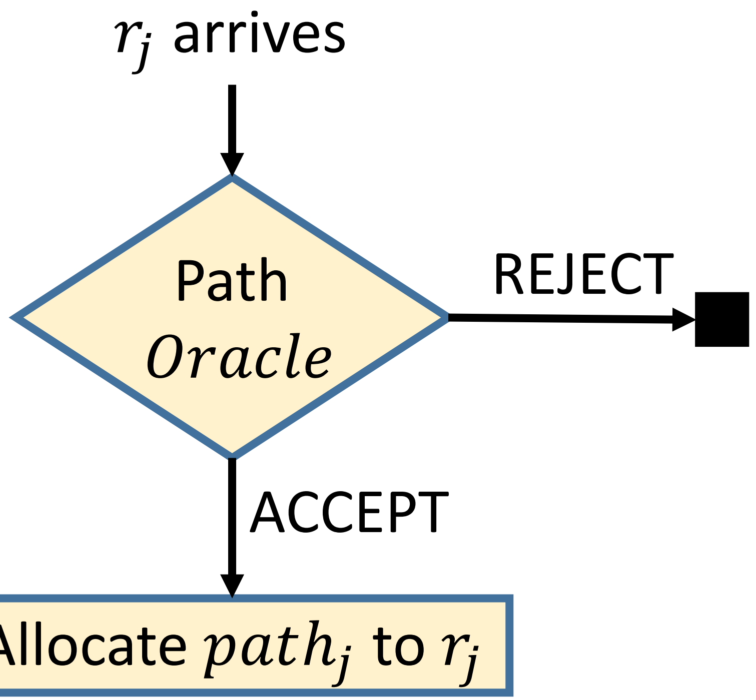
- ACC/REJECT $= \sum_{accepted} b_j$
- ACC/STDBY$= \sum_t \sum_{accepted(t)} b_j$

- Known duration ≈ only arrivals (persistent requests)

Online Persistent Routing Algorithm [AAP93]

(known duration, ACC/REJ)

- State: accepted requests $\{r_j, path_j\}_{acc.}$
- r_j arrives:



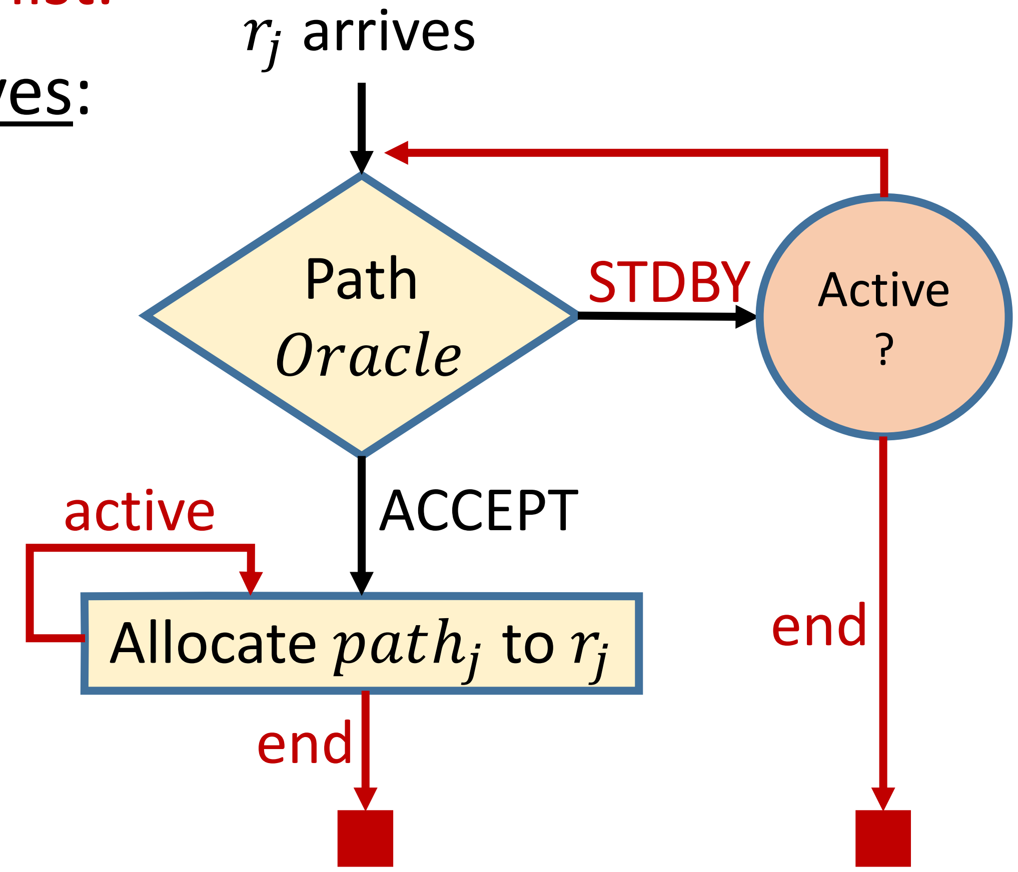
- $Oracle(r_j)$
  - $w(e) = \exp(load(e))$
  - Search for a path $s_j \rightsquigarrow t_j$ s.t. $\sum_{e \in path_j} w(e) < b_j$
  - Success: **return**($\boldsymbol{path_j}$)
  - Fail: **REJECT**

$load(e) \triangleq flow(e)/c(e)$

Our Algorithm

(unknown duration, ACC/STDBY)

- State: accepted requests $\{r_j, path_j\}_{acc.}$, STDBY list.
- r_j arrives:



- $Oracle(r_j)$
  - $w(e) = \exp(load(e))$
  - Search for a path $s_j \rightsquigarrow t_j$ s.t. $\sum_{e \in path_j} w(e) < b_j$
  - Success: **return($path_j$)**
  - Fail: **STDBY**

$load(e) \triangleq flow(e)/c(e)$

Main Result (virtual circuit routing version)

Thm.

Small demands

$$\Longrightarrow$$

in each time step t,

$$benefit_t(ALG) \geq \frac{benefit_t(OPT)}{\log(|V|b_{\max})}.$$

Remarks:

1. Small demands: $d_{\max} \leq \frac{c_{\min}}{\log(|V|b_{\max})}$.
2. If served, request is served **continuously** until it ends.
3. Even when $\boldsymbol{OPT}$ is **fractional**, may **preempt**, may **reroute**, i.e., MCF in every time step.
4. Proof idea: repeating the “game” of persistent routing.
5. Asymptotically optimal!
6. Extends to the SDN modeling (next slide).

Modeling SDN

- Network $N = (V, E)$
  - V set of servers
  - E set of links
  - $c: V \cup E \to \mathbb{N}^+$
    - i.e, Processing, BW
- Online requests $r_j = \langle G_j, d_j, b_j, U_j \rangle$
  - $G_j = (X_j, Y_j)$ PR-graph
  - $d_j: X_j \cup Y_j \to \mathbb{N}$ demand
  - $b_j \in \mathbb{N}^+$: benefit per served time step
  - $U_j: X_j \cup Y_j \to 2^V \cup 2^E$ mapping

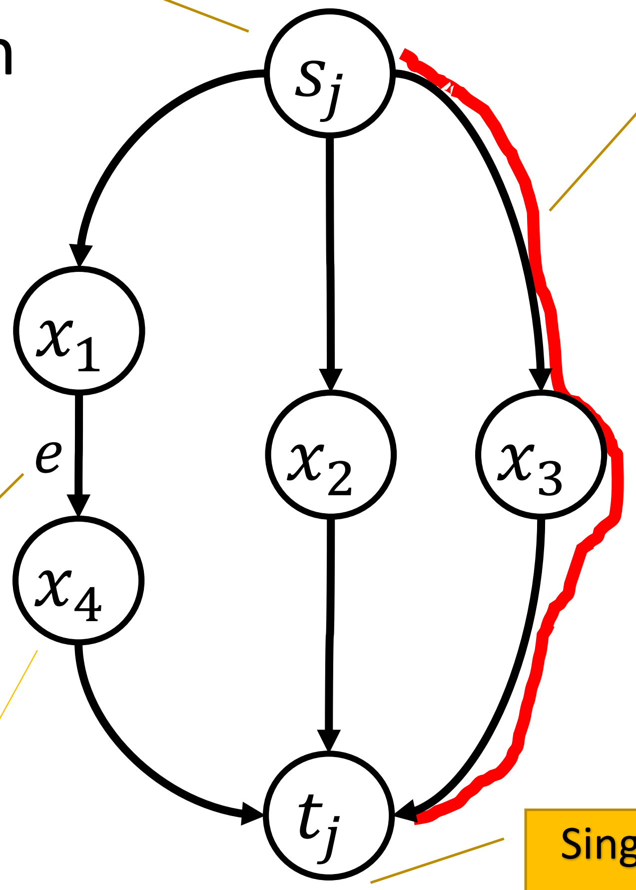
- PR-graph

Single source – packets arrive from here

Req served by any $s - t$ path. Needs mapping to N

Can be implemented by $U(x_1) = \{v_{17}, v_{23}, v_{97}, \dots\}$

e can be implemented by $v_0 \xrightarrow{p} v_k$ if $\forall i \ (v_{i-1}, v_i) \in U(e)$



Action vertices – enc/dec, comp/decomp…

Single sink – packets are destined to here

Examples

Simple Routing

"I want to route a connection of 100 mbps from $\boldsymbol{v}$ to $\boldsymbol{v}'$ "

- PR-graph $s \xrightarrow{e} t$
- $U(s) = \{v\}$,
- $U(t) = \{v'\}$,
- $U(e) = E$,
- $d(e) = 120\ mbps$

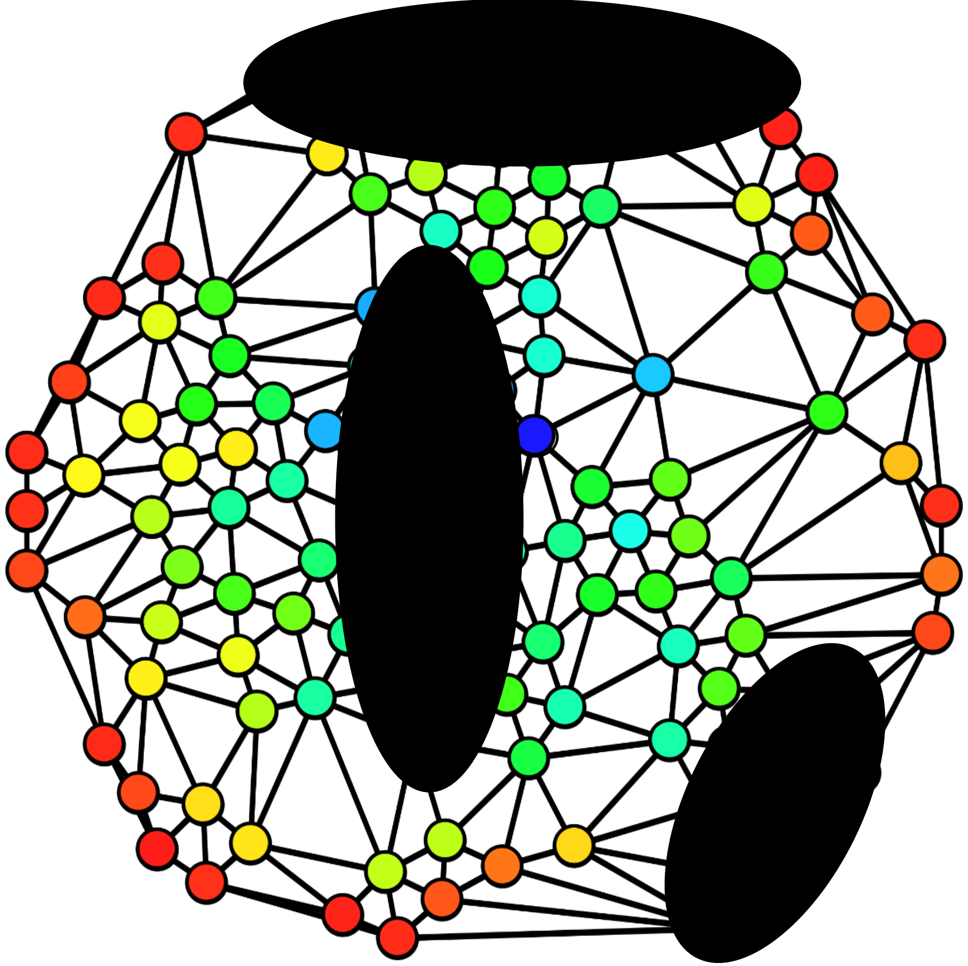
Serial Processing

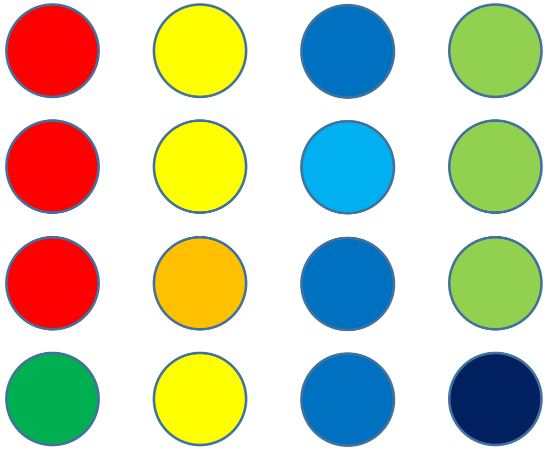
- Stream,
- Pass k transformations $a_1, \dots, a_k$ in series.

- PR-graph $s \to a_1 \to \cdots \to a_k \to t$
- $U(a_i) \subseteq V$ implements a_i
- Can model BW changes
  - E.g., a_j is COMPRESS.
  - How? Set diff demands to diff PR-edges.

Summary

- Need to **allocate** network resources in **online** fashion.
- Each request specifies sequence(s) of **functions**.
- **Map** desired functions to **real** network devices.
- **Route** the request between these devices.
- **Security**: want to **avoid** some **nodes.**
- **Flexibility**: customer does not want to commit to the duration upfront ➔ the **duration is unknown**.
- **Maximum benefit**: maximizing the benefit rate.






That's all Folks!