Scalable, Optimal Flow Routing in Datacenters via Local Link Balancing

Flow Routing in Datacenter

- Several exsiting proposals includes: ECMP, MPTCP, centralized sheduler and switch local schemes.
- Routing flows in a capacitated network is a multi-commodity flow problem.
- The optimal solution of the flow routing problem is achived through flow spliting.
- Flow spliting causes packet reordering. This could degrade TCP performance.

The Architecture of Local Flow

- Local flow splits flows to achieve optimal performance.
- A similar idea called packet-scatter also splits flows.
- But packet-scatter splits every individual flows to each out-going links.
- Local flow tries to distribute all the flows with the same destination to each outgoing links and only splits flows when necessary.

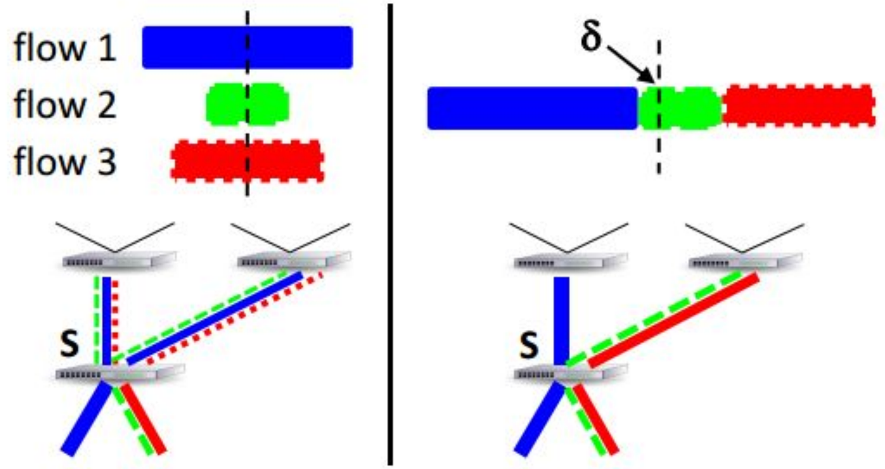


Figure 1: A set of flows to the same destination arrives at switch S. PacketScatter (left) splits every flow, whereas LocalFlow (right) distributes the aggregate flow, and only splits an individual flow if the load imbalance exceeds δ.

The Architecture of Local Flow

- Main LocalFlow sheduling loop:
  - 1. Measure the rate of each active flow. This is done by querying the byte counter of each forwarding rule from the previous interval and dividing by the interval length.
  - 2. Run the scheduling algorithm using the flow rates from step 1 as input.
  - 3. Updates the rules in the forwarding table based on the outcome of step 2, and reset all byte counters.

- LocalFlow utilizes some features of OpenFlow.

Architecture of Local Flow

- LocalFlow only achieves optimal throughput on networks with symmetry property.
- Sysmetry property: A network is symmetric if for all source destination pairs (s, d), all switches on the shortest paths between s and d that are the same distance from s have identical outgoing capacity to d.

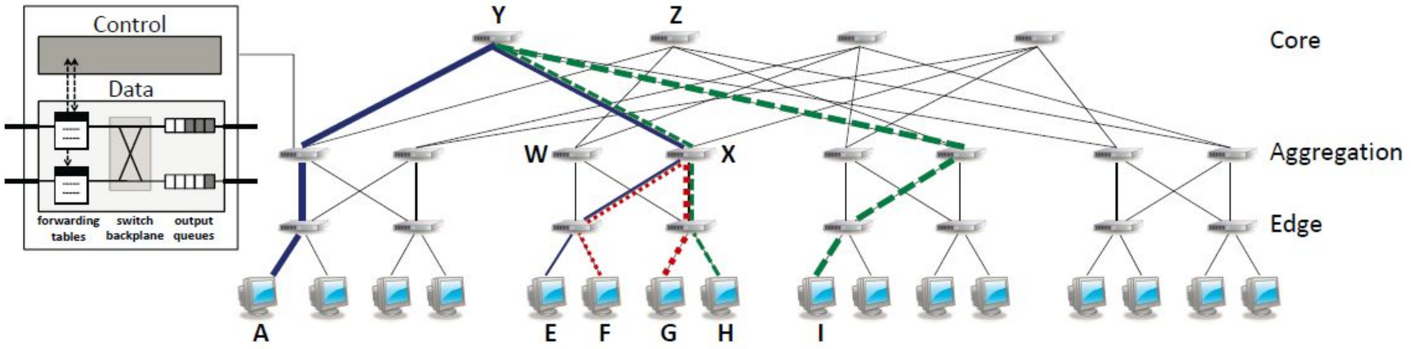


Figure 2: A FatTree network with 4-port switches. VL2 is a variation on this topology. End hosts A, G, I simultaneously transmit to E, F, H and collide at switches Y and X, but there is sufficient capacity to route all flows at full rate.

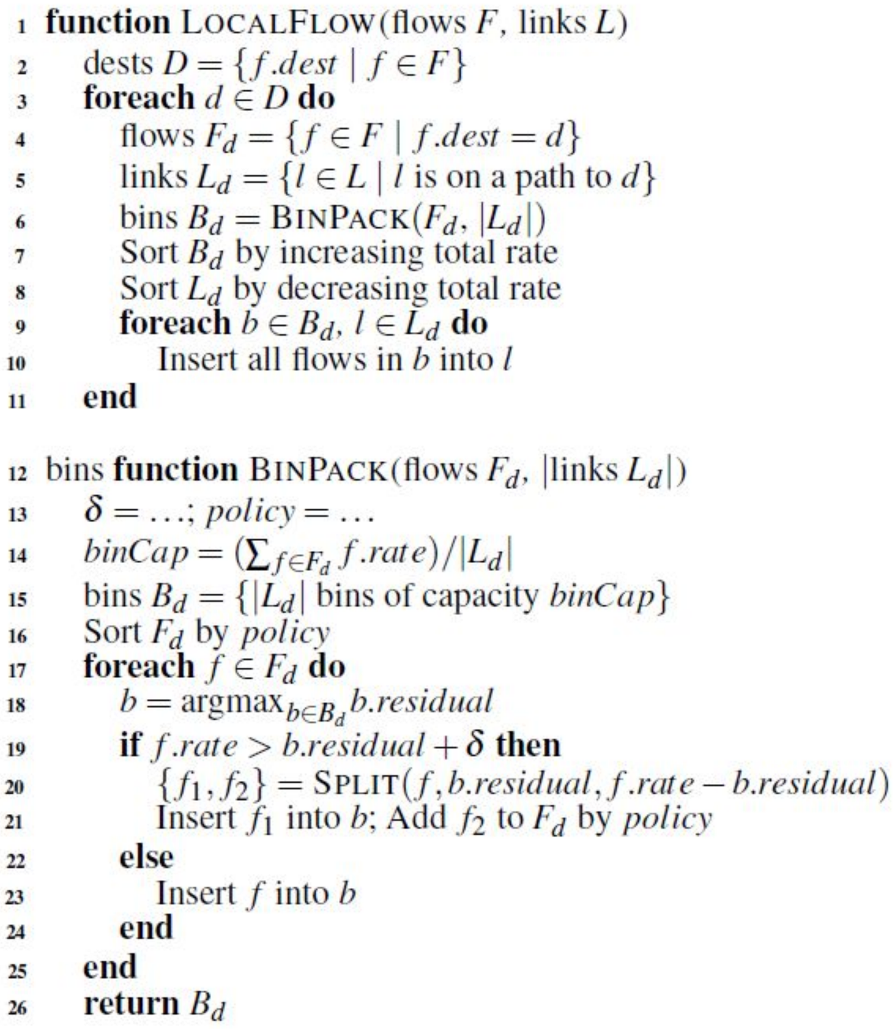
Architecture of Local Flow

- The key purpose of Local Flow is to find optimal flow routing for the following maximum MCF problem:

$$
\begin{aligned}
&\text{maximze}: \sum_i U_i(x_i) \\
&\text{subject to}: \sum_{u:(u,v)\in E} f_{u,v}^{s,d} = \sum_{w:(v,w)\in E} f_{v,w}^{s,d} : \forall v, s, d, \\
&\qquad \sum_{u:(s,u)\in E} f_{s,u}^{s,d} = \sum_{i:s\to d} x_i : \forall s, d \\
&\qquad \sum_{s,d} f_{u,v}^{s,d} \le c_{u,v} : \forall (u,v) \in E, \text{ link capacity } c_{u,v}
\end{aligned}
$$

- Local Flow achieve maximum MCF objective in the following way:
- First: In each switch, distribute flows with the same destination evenly among the out-going links. This generates a stable routing matrix and balanced links.
- Second: Rely on the TCP protocol of each end host to maximze the MCF objective. As the duality model of TCP indicates that TCP will maximze the total network utility when giving a fixed routing matrix.

Architecture of Local Flow

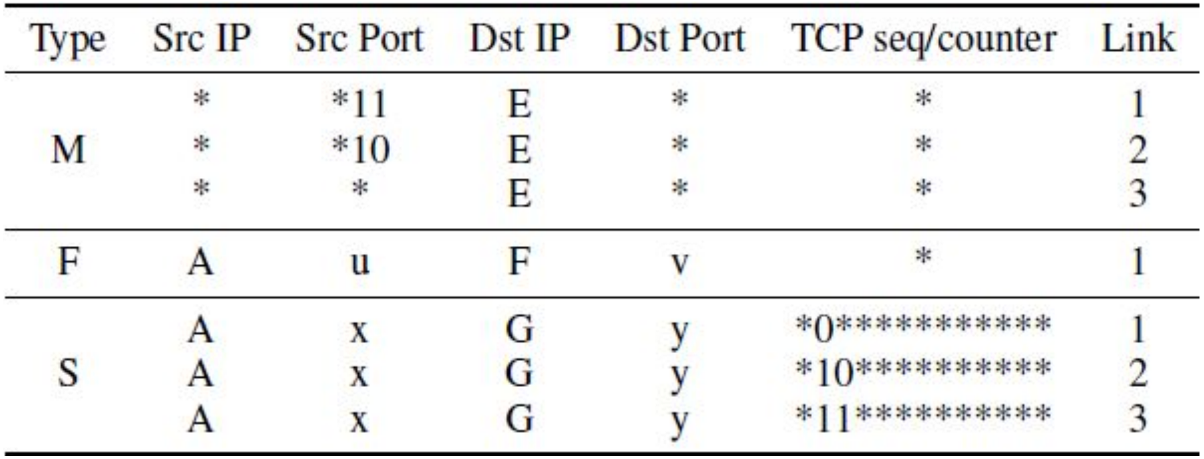


```
function LocalFlow(flows $F$, links $L$)
    dests $D = \{f.dest \mid f \in F\}$
    foreach $d \in D$ do
        flows $F_d = \{f \in F \mid f.dest = d\}$
        links $L_d = \{l \in L \mid l$ is on a path to $d\}$
        bins $B_d = \text{BinPack}(F_d, |L_d|)$
        Sort $B_d$ by increasing total rate
        Sort $L_d$ by decreasing total rate
        foreach $b \in B_d, l \in L_d$ do
            Insert all flows in $b$ into $l$
    end

bins function BinPack(flows $F_d$, |links $L_d$|)
    $\delta = \ldots$; $policy = \ldots$
    $binCap = (\sum_{f \in F_d} f.rate)/|L_d|$
    bins $B_d = \{|L_d|$ bins of capacity $binCap\}$
    Sort $F_d$ by $policy$
    foreach $f \in F_d$ do
        $b = \text{argmax}_{b \in B_d} b.residual$
        if $f.rate > b.residual + \delta$ then
            $\{f_1, f_2\} = \text{Split}(f, b.residual, f.rate - b.residual)$
            Insert $f_1$ into $b$; Add $f_2$ to $F_d$ by $policy$
        else
            Insert $f$ into $b$
        end
    end
    return $B_d$
```

Architecture of Local Flow

- How to implement the split function in the Local Flow algorithm? The author introduces multi-resolution splitting. It uses some advanced features in the new Openflow specification.
- A single flow is represented using flow's 5-tuple.
- Flows are devided into flowlets. Each flowlet is several contiguous TCP packets of t bytes. Flowlets are grouped into subflows.
- Subflows are matched using the TCP sequence number.



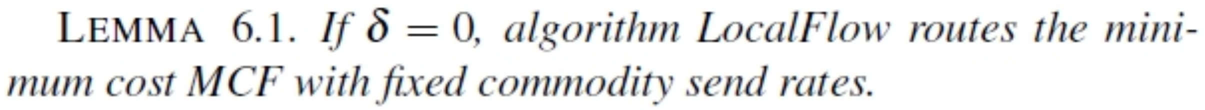
Type	Src IP	Src Port	Dst IP	Dst Port	TCP seq/counter	Link
M	*	*11	E	*	*	1
	*	*10	E	*	*	2
	*	*	E	*	*	3
F	A	u	F	v	*	1
S	A	x	G	y	*0***********	1
	A	x	G	y	*10**********	2
	A	x	G	y	*11**********	3

Analysis

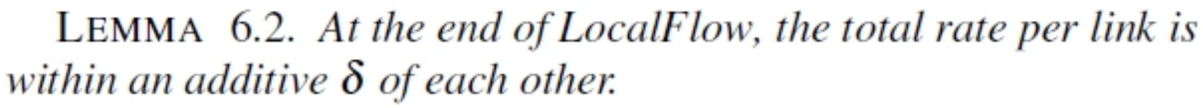
- The ‘master’ LocalFlow optimization adapts link flow rates $f_{u,v}^{d}$ to minimize the maximum link cost , $\frac{\sum_d f_{u,v}^{d}}{c_{u,v}}$ for the commodity send rates x_i^* determined by the slave TCP sub-problem.

- The link-balanced flow rates determined by LocalFlow lead to an optimal solution to the original MCF objective.

Analysis



LEMMA 6.1. *If $\delta = 0$, algorithm LocalFlow routes the minimum cost MCF with fixed commodity send rates.*



LEMMA 6.2. *At the end of LocalFlow, the total rate per link is within an additive δ of each other.*

- These two lemmas demonstrate that the LocalFlow algorithm achive link-balanced flow rates on each link.

Analysis

- In order to show that the link-balalnced flow rates enable TCP to maximize the MCF objective, the NUM formulation of the TCP sub-problem is considered.

$$\text{maximize}: \sum_i U_i(x_i)$$

$$\sum_i x_i \sum_{p:(u,v)\in p} \pi_i^p \le c_{u,v} : \forall(u,v) \in E$$

$$\sum_p \pi_i^p = 1 : \forall i$$

- Here the π_i^p is the path probability.It determines the proportion of commodity x_i's traffic that traverses path p.

Analysis

- The optimality condition for the TCP sub-problem is solved using dual decomposition.
- Form the Lagrangian function $L(x,\lambda)$ by introducing dual variable $\lambda_{u,v}$ for each link. $\lambda_{u,v}$ is an indicator of congesiton on each link.

$$L(x,\lambda)=\sum_i\int f(x_i)dx_i-\sum_{u,v}\lambda_{u,v}(\sum_i x_i\sum_{p:(u,v)\in p}\pi_i^p-c_{u,v})$$

- Next construct the Lagrange dual function $Q(\lambda)$ maximized with respect to x_i :

$$x_i^*=f^{-1}(\beta):\text{when }\frac{\partial L}{\partial x_i}=0,\ \forall x_i,\ \beta=\sum_p\pi_i^p\sum_{(u,v)\in p}\lambda_{u,v}$$

$$Q(\lambda)=\sum_i(\int f(x_i^*)dx_i^*-f^{-1}(\beta)\beta)+\sum_{u,v}\lambda_{u,v}\cdot c_{u,v}$$

Analysis

- Minimizing $Q(\lambda)$ with respect to λ gives both the optimal dual and primal variables, since the original objective is concave.

$$\sum_i f^{-1}(\beta) \sum_{p:(u,v)\in p} \pi_i^p = c_{u,v}, \text{ when } \frac{\partial Q}{\partial \lambda_{u,v}} = 0, \forall (u,v) \in E$$

- When this condition is satisfied, the network system reaches maximum network utility. TCP computes this solution in a distributed fashion using gradient ascent.

Analysis

- According to symmetry property, for all links from a node u to nodes (v,w) in the next level of a shortest path tree:
  - The links have the same capacity, thus we have:

$$c_{u,v} = c_{u,w}$$

  - When the system reaches maximum network utility, we have:

$$\sum_{i} f^{-1}(\beta) \sum_{p:(u,v)\in p} \pi_i^p = \sum_{i} f^{-1}(\beta) \sum_{p:(u,w)\in p} \pi_i^p$$

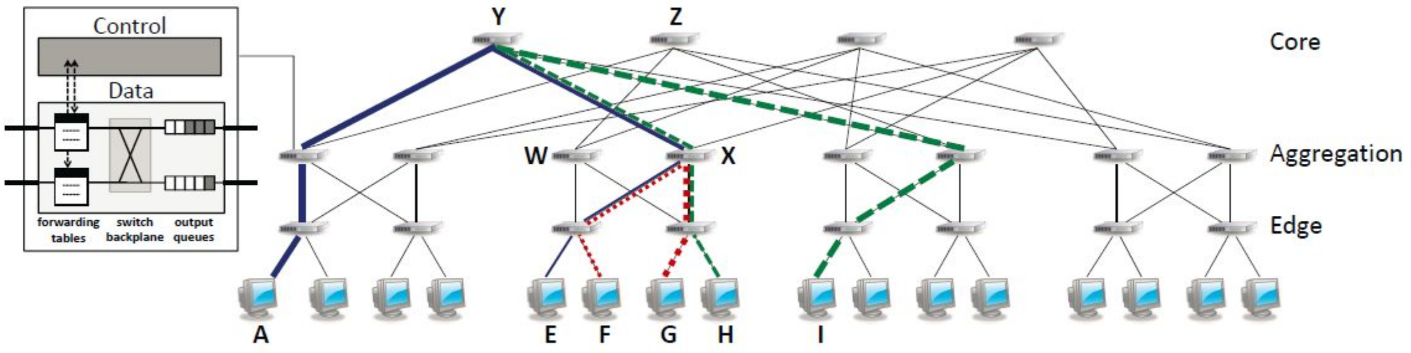


Figure 2: A FatTree network with 4-port switches. VL2 is a variation on this topology. End hosts A, G, I simultaneously transmit to E, F, H and collide at switches Y and X, but there is sufficient capacity to route all flows at full rate.

Analysis

- LocalFlow balance the flow rates across links for flows to the same destination. When the maximum network utility is reached, the condition:

$$\sum_{i} f^{-1}(\beta) \sum_{p:(u,v)\in p} \pi_i^p = \sum_{i} f^{-1}(\beta) \sum_{p:(u,w)\in p} \pi_i^p$$

- Become:

$$\sum_{i:s\to d} f^{-1}(\beta) \sum_{p:(u,v)\in p} \pi_i^p = \sum_{i:s'\to d} f^{-1}(\beta) \sum_{p:(u,w)\in p} \pi_i^p$$

- Note that LocalFlow will adjust flow rates to achieve the above goal in response to TCP's optimized sending rates.

Analysis

- LocalFlow minimize the maximum link utilization by balancing per-destination link flow rates, opening up additional head room on each link for the current commodity send-rates x_i^* to grow.

- Between the LocalFlow operation interval, the TCP maximizes its send rate objective to consume the additional capacity.

- Given proper time scale, the distributed convex optimization process converges to an optimal network utility.

Evaluation

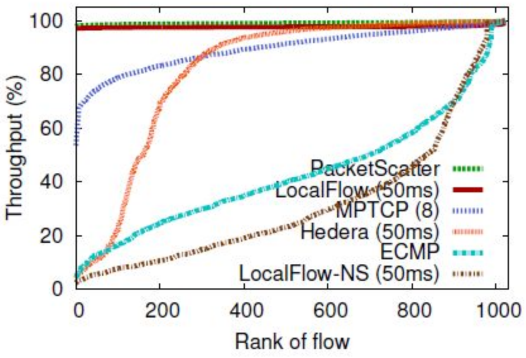


Figure 4: Individual flow throughputs for a random permutation on a 1024-host Fat-Tree network.

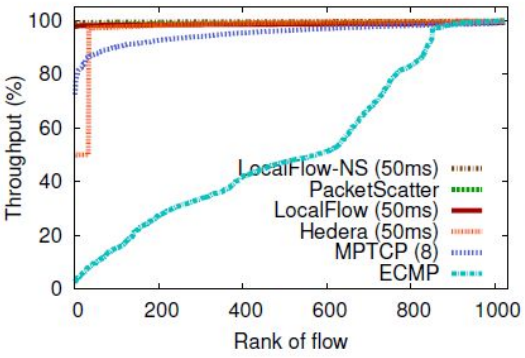


Figure 5: Individual flow throughputs for a stride permutation on a 1024-host Fat-Tree network.

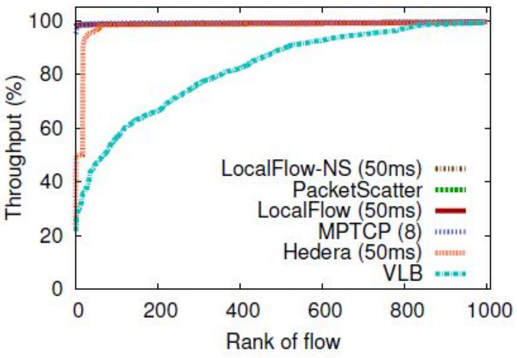
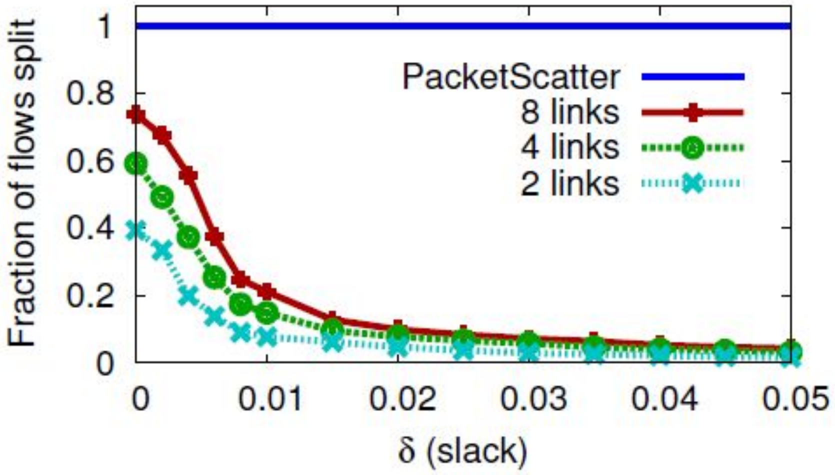


Figure 6: Individual flow throughputs for a random permutation on a 1000-host VL2 network.

Evaluation



Avg. # of subflows/flow	2 links	4 links	8 links
LocalFlow ($\delta = 0$)	1.35	2.08	3.54
LocalFlow ($\delta = 0.01$)	1.06	1.20	1.49
LocalFlow ($\delta = 0.05$)	1.01	1.04	1.09
PacketScatter	2	4	8

Figure 9: Fraction of flows split (top) and average number of subflows per flow (bottom) by LocalFlow for different numbers of outgoing links, compared to other protocols, using a 3914-second trace from a real datacenter switch.