Valiant's Routing

Juan Vanerio

Communication Technologies Group
Faculty of Computer Science
University of Vienna

December 2, 2020

Table of Contents

Introduction

- **Problem: Communications on a distributed system.**
- Message forwarding is the lowest level operation.
- Oblivious routing allows for no costly central control.
- Deterministic forwarding results in congestion w.h.p.
- Valiant's routing (1981) transforms a deterministic forwarding in two random forwardings in tandem.

Why is it (still) relevant...?

- **Fast and efficient communication on a distributed system.**
- Solves the case of (partial) permutations.
- Provides:
  - Speed: $O(logN)$ to complete permutation.
  - Low overhead: $O(logN)$ bits
- Supports unpredictable traffic.
- Uses mostly static routing (stable).

Why is it (still) relevant...?

Has been adapted to many topologies:

- Full Mesh: each phase involves only one hop.
- Fat Tree [3]
- Dragonfly [11]
- In general by sending packets through a randomly selected intermediate node out of all nodes in the datacenter network [8].

Adapted also for...

- Internet backbone, ISP networks, VPN services and Autonomous systems[12][5].
- Data Center Networks [3].
- Switching fabric of a packet switch [2].
- Traffic flows instead of packets. [3][7].

Problem Statement

- Given:
  - A sparsely connected distributed system with N nodes.
  - Each node has a packet to send to other node.[1]
  - No packet has same destination as another.[1]
  - Nodes can relay packets at discrete time steps.
  - At most one packet per edge at a time.
- Find an algorithm that forwards the packets correctly and finishes quickly (within time $O(logN)$)

[1] Some conditions have been relaxed in further works.

Considerations

- **Hypercube network**
  - Hypercube with $N = 2^n$ nodes
  - $Nn/2$ bidirectional edges (or Nn directed edges)
  - Binary representation
    - Node $x \Rightarrow x_1 x_2 \ldots x_{i-1} x_i x_{i+1} \ldots x_n$ with $x_i \in \{0, 1\}, i \in [N]$
  - Hamming distance $H(u, v) = \sum_{i=1}^{N} (u \, XOR \, v)_i$
  - $neigh(v) = \{u | H(u, v) = 1\}$
  - $|neigh(v)| = n$
- Output queue per interface.
- Permutation:
  - Full: $d : [N] \rightarrow [N]$ $(\forall s \in [N], d(s) \in [N])$
  - Partial: $d : U \rightarrow V, s.tU, V \subseteq [N], |U| = |V|$
  - Bijective

Definitions

- **Route**: sequence of edges for a packet to get from source to destination.
  - Each dimension might be traversed only once.
- **Collision**: 2 or more packets arrive at the same node at the same time step and are allocated to the same outgoing interface.
- **Congestion**: Presence or number of queued packets.
  - Introduces delays.

Bit Fixing (I)

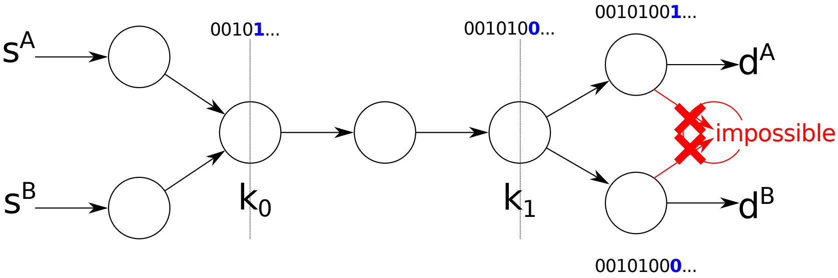
- Deterministic forwarding algorithm for the hypercube
- Node $s \Rightarrow s_1 \ldots s_n$ has a packet to send to node $d(s) \Rightarrow d_1 \ldots d_n$.
  - Start from first bit (MSB)
  - For each bit (dimension) i, fix if $s_i \neq d_i$
- Optimal for forwarding a single packet:
  - Changes only bits on which s and d differ.
  - Minimal distance.

Bit Fixing (II)

Claim 1

Two routes with the same bit-fixing scheme can only intersect in a consecutive sequence of edges.

aka: Two packets may come together along a route segment and then separate, but only once.



Bit Fixing (III)

Proof.

- Given two packets p^A and p^B colliding at for the first time at step k_0 and then traversing the same edges until some step k_1:
  - Let $p^A(p^B)$ have source $s^A(s^B)$ and destination $d^A(d^B)$
- Then at any step $l \in [k_0, k_1]$:
  - Destinations equal up to bit l: $d_1^A \dots d_l^A = d_1^B \dots d_l^B$
  - Sources equal in last $n - l$ bits: $s_{l+1}^A \dots s_{l+1}^A = s_n^B \dots s_n^B$
- On step $k_1 + 1$, the routes separate, so $d_{k_1+1}^A \neq d_{k_1+1}^B$
- If the two routes collide at some step $k_2 > k_1$, then they should be identical up to bit k_2.
- Then they can overlap only once. □

Deterministic Routing

Theorem 1

Any deterministic oblivious permutation routing scheme for a parallel machine with N processors, each with n outward links requires $\Omega\left(\sqrt{\frac{N}{n}}\right)$ steps.

- For proof, see [4]
- So, congestion whp...
- Worsens when n grows.

An example

We now provide an example with bit-fixing.

Examples

- Assume n is even, and write $x \in [N]$ as $x = (x', x'')$ with $x', x'' \in \{0,1\}^{\frac{n}{2}}$
- Consider any permutation π which maps $(x', 0)$ to $(0, x'')$
- Notice that these $2^{n/2} = \sqrt{N}$ packets must go through node $(0,0)$
- We will need $\frac{\sqrt{N}}{n}$ steps to send them through.

Valiant's Routing

Use randomized routing to reduce congestion!

- Instead of sending packets directly from their source s to their destination $d(s)$, send them through a random intermediate node.

Valiant's Routing

Use randomized routing to reduce congestion!

- Instead of sending packets directly from their source s to their destination $d(s)$, send them through a random intermediate node.
- Split the forwarding in two phases in tandem:
  - **Phase A**:
    - Bit-fix forward all the packets using a random permutation $\pi_A : [N] \to [N]$
    - $\forall s, s' \in [N]\,,\ Pr(s' = \pi_A(s)) = \frac{1}{N}$
  - **Phase B**: Bit-fix forward all packets from $\pi_A(s)$ to $d(s)$.

Valiant's Routing

Use randomized routing to reduce congestion!

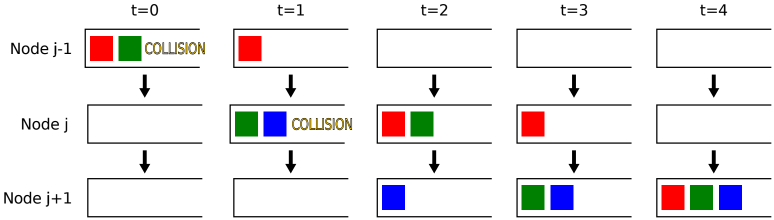
- Instead of sending packets directly from their source s to their destination $d(s)$, send them through a random intermediate node.
- Split the forwarding in two phases in tandem:
  - **Phase A**:
    - Bit-fix forward all the packets using a random permutation $\pi_A : [N] \to [N]$
    - $\forall s, s' \in [N]\,,\ Pr(s' = \pi_A(s)) = \dfrac{1}{N}$
  - **Phase B**: Bit-fix forward all packets from $\pi_A(s)$ to $d(s)$.
- Phase A (B) is a permutation from given (random) sources to random (given) destinations.
- Phase B can be thought of as Phase A played in reverse.
  - Proofs for Phase A are then analogous for Phase B.

Bound for Maximum Delay

- Let R_i be the route of packet sourced at node $i \in [N]$ and Δ_i its queuing delay.
- Total delay from i to $\pi_A(i)$ is $\tau_i^A = O(n + \Delta_i)$, propagation + queuing delay.

Theorem 2

$\Delta_i \leq |S_i|$ with S_i the set of packets whose routes intersect R_i.



Bound for Maximum Delay

Proof outline:

- No collisions imply no additional delay.
- On collisions, one packet gets delayed by another just once.
  - On the segment they share, the packets won't be again on the same node unless the first of them has a new collision with a third packet.
- Recall from Claim 1 that routes may overlap only once so collisions between any two packets may happen just once.
- Each intersecting route may cause only one collision, then adding just one extra delay.

Symmetry

Symmetric scheme: if for any two edges, the expected number of routes that go through each one of them is the same.

Claim 2

Valiant's is a symmetric scheme with $\frac{1}{2}$ routes through each edge on each direction.

Proof:

- Let $T(e)$ be the number of routes that pass through edge e.
- Write e as $e = (u, v)$ s.t. $u_i \neq v_i$
- Any source/packet x that may reach u not from v satisfies (1) $x_j = u_j, j \in [i, n]$ and (2) $u_j = \pi_{A,j}(x)$ for $j \in [1, i-1]$.
  - (1) $\Rightarrow |\{x\}| = 2^{i-1}$
  - (2) and $Pr\left(\pi_A(x)\right) = \frac{1}{N} \Rightarrow P_u = Pr\left(x\, in\, u\right) = 2^{-(i-1)}$
- For each x, $P_{(u,v)} = Pr\left(x\, through\left(u, v\right) | \, x\, in\, u\right) = \frac{1}{2}$
- $E\left[T(e)\right] = \sum_{k=1}^{N} Pr(R_k\, through\, e) = \sum_{\{x\}} P_u P_{(u,v)} = \frac{1}{2}$ □

Valiant's theorem

Main Theorem

With probability at least $1 - 2^{-(C-\frac{3}{2})n}$ every packet reaches its intermediate destination $\pi_A(i)$ in $(C+1)n$ or fewer steps.

Proof:

- Let $H_{ij} = \begin{cases} 1 & \text{if } |R_i \cap R_j| \geq 1 \\ 0 & \text{else.} \end{cases}$ and $T(e)$ the number of routes through edge e.
- See that $|S_i| = \sum_{j=1}^{N} H_{ij}$
- Say $R_i = (e_1, \ldots, e_k)$, then $\sum_{j=1}^{N} H_{ij} \leq \sum_{l=1}^{k} T(e_l)$
- Merging, taking expectations and bounding again:

$$E\left[|S_i|\right] \leq \sum_{l=1}^{k} E\left[T(e_l)\right]$$

Valiant's theorem

- By Claim 2, $E\left[T(e_l)\right] = \frac{1}{2}$ then:

$$E\left[|S_i|\right] \leq \frac{k}{2} \leq \frac{n}{2}$$

- As $|S_i|$ is a sum of binary variables, we can use a Chernoff bound for $\delta > 2e - 1$ to get:

$$Pr\left(|S_i| > (1+\delta)\mu\right) \leq \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\mu} < 2^{-\delta\mu}$$

- Consider a large enough C s.t. $(1+\delta)\mu = Cn$. As $\mu \leq n/2$, it follows that

$$\delta\mu \geq (C - \frac{1}{2})n$$

- Then

$$Pr\left(|S_i| > Cn\right) < 2^{-(C-\frac{1}{2})n} \;\; \forall i \in [N]$$

Valiant's Theorem

- Let E_i be the event defined by $|S_i| > Cn$
- Theorem 2 states $\Delta_i \leq |S_i|$, so $\neg E_i$ implies $\Delta_i \leq Cn$.
- The probability that *no packet* gets delayed more than Cn steps can then be bounded by:

$$\begin{aligned} Pr\left(\text{no packet has delay} > Cn\right) &\geq 1 - Pr\left(\bigcup_{i=1}^{N} E_i\right) \\ &\geq 1 - \sum_{i=1}^{N} Pr\left(E_i\right) \\ &\geq 1 - 2^n 2^{-(C-\frac{1}{2})n} = 1 - 2^{-(C-\frac{3}{2})n} \end{aligned}$$

- Finally recall that the time to finish a phase is bounded by its queuing delay plus n, required to transmit the messages. □

Valiant's theorem

- Valiant also proved that bit fixing is not necessary [10].
- So the result holds for routes chosen at random order of dimensions.
- Although the proofs are harder.
- Let F (G) be the max duration of Phase A (B).

Valiant's Original Theorem

For each constant S, there exists C, $F = G = Cn$, s.t. with probability at least $1 - 2^{-Sn}$ every phase completes.

References

[1] John Canny. *CS174 Lecture 11*. URL: https://people.eecs.berkeley.edu/~jfc/cs174/lecs/lec11/lec11.pdf. (accessed: 12.11.2020).

[2] Cheng-Shang Chang, Duan-Shin Lee, and Yi-Shean Jou. "Load balanced Birkhoff–von Neumann switches, part I: One-stage buffering". In: *Computer Communications* 25.6 (2002), pp. 611–622.

[3] Albert Greenberg et al. "VL2: a scalable and flexible data center network". In: *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 2009, pp. 51–62.

[4] Christos Kaklamanis, Danny Krizanc, and Thanasis Tsantilas. "Tight bounds for oblivious routing in the hypercube". In: *Mathematical Systems Theory* 24.1 (1991), pp. 223–232. DOI: 10.1007/BF02090400.

References (cont.)

[5] Murali Kodialam, TV Lakshman, and Sudipta Sengupta. "Efficient and robust routing of highly variable traffic". In: *In Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III*. Citeseer. 2004.

[6] Shachar Lovett. *CSE 190, Great ideas in algorithms:Randomized routing*. URL: http://cseweb.ucsd.edu/~slovett/teaching/SP15-CSE190/randomized_routing.pdf. (accessed: 13.11.2020).

[7] Santosh Mahapatra and Xin Yuan. "Load balancing mechanisms in data center networks". In: *the 7th int. conf. & expo on emerging technologies for a smarter world (cewit)*. 2010.

[8] N. Maksic and A. Smiljanic. "Improving utilization of data center networks". In: *IEEE Communications Magazine* 51.11 (2013), pp. 32–38. DOI: 10.1109/MCOM.2013.6658649.

References (cont.)

[9] L. G. Valiant and G. J. Brebner. "Universal Schemes for Parallel Communication". In: *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*. STOC '81. Milwaukee, Wisconsin, USA: Association for Computing Machinery, 1981, pp. 263–277. ISBN: 9781450373920. DOI: 10.1145/800076.802479. URL: https://doi.org/10.1145/800076.802479.

[10] Leslie G. Valiant. "A scheme for fast parallel communication". In: *SIAM journal on computing* 11.2 (1982), pp. 350–361.

[11] Timothy Yuan et al. "HyperOXN: A Novel Data Center Topology Driven by Machine Learning". In: *2018 10th International Conference on Communication Software and Networks (ICCSN)*. IEEE. 2018, pp. 573–578.

References (cont.)

[12] R. Zhang-Shen and N. McKeown. "Designing a Fault-Tolerant Network Using Valiant Load-Balancing". In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. 2008, pp. 2360–2368. DOI: 10.1109/INFOCOM.2008.305.

Thank You

Thank You for listening!
Questions?

Why is Phase A necessary?

Is it enough to randomly pick between the many shortest paths from s to $d(s)$?

- Intuitively, that may not be enough entropy to generate exponential probabilities.
  Argument:
- Assume $N = 2^n$ is large enough, divisible by 4 and let $r = N/4$.
- Consider edge $e = (x, y)$ with $x_i \neq y_i$.
- Let $W^{(r)} = \{w | H(w, x) = r, w_i = x_i\}$ and
  $Z^{(r)} = \{z | H(z, y) = r, z_i = y_i, H(z, w) = 2r + 1 \forall w \in W^{(r)}\}$
- Routes R that may go from w to z through e: $|W^{(r)}| = |Z^{(r)}| = \dfrac{(n-1)!}{r!(n-r-1)!}$
- Routes from w to z: $(2r + 1)!$
- Routes from w to x (or from y to z): $r!$
- Then, $Pr(R \cap e) = \dfrac{(r!)^2}{(2r+1)!}$

Why is Phase A necessary? (cont.)

- Consider a permutation $W^{(r)} \to Z^{(r)}$. Then, the expected number of routes through e is

$$
\begin{aligned}
|W^{(r)}|Pr(R \cap e) &= \frac{(4r-1)!r!}{(3r-1)!(2r+1)!} \\
&= \frac{(4r-1)\dots(3r)}{(2r+1)\dots(r+1)} \\
&\geq \frac{1}{3r}\left(\frac{3r}{2r+1}\right)^{r} \\
&\geq N^{\gamma}
\end{aligned}
$$

  for large enough n, with $0 < \gamma < \frac{1}{4}\log_2 \frac{3}{2}$.

- Then the number of routes through edge e is bounded below by a potential function of N.