Optimal Network Flow Allocation

EE 384Y
Almir Mutapcic and Primoz Skraba
27/05/2004
Problem Statement

- Optimal network flow allocation
  - Find flow allocation which minimizes certain performance criterion
    - Lowest average delay through the network
    - Minimize maximum link utilization
    - Fair bandwidth allocation and QoS agreements
  - Trade-off between optimality and simplicity
  - Devise practical schemes with low computational complexity and guaranteed performance bounds
Motivation

- Internet backbone and PoPs are over-engineered
  - Overcome link failures
  - Underutilized (multiple routes exist)
- Current Protocols
  - Typically find shortest path(s)
  - Do not directly minimize delay through PoPs
Background

- IS-IS & OSPF
  - Link-state routing protocols
  - Limited load balancing
  - Manually tuned to a few routes (traffic engineering)

- MPLS
  - Re-labels packets in the internal network

- Previous work
  - Resource Allocation (minimize max link utilization)
  - Routing Heuristics
Informal Formulation

- Optimal network flow allocation
  - Decide how to distribute packets from a particular flow across the network links ($x$ variables)
  - Satisfy conservation laws

- Definition of a flow
  - Aggregate flow to each destination (sink) node
  - Every other node can be a source to the sink node
  - Source-sink vector

$$s_d^{(d)} = - \sum_{i \in \mathcal{N}, i \neq d} s_i^{(d)}$$
Mathematical Formulation

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{L} \phi_i (T_i) \\
\text{subject to} & \quad Ax^{(d)} = s^{(d)} \\
& \quad x^{(d)} \geq 0 \\
& \quad T \leq C, \\
\end{align*}
\]

where \( d = \{1, \ldots, D\} \).

- Convex cost function for each link \( i \)
- Total link \( i \) traffic \( T_i = \sum_{d \in D} x_i^{(d)} \) (\( x \) is flow’s traffic)
- Node-link incidence matrix – \( A \)
- Link capacity vector – \( C \)
Piece-wise Linear Approximation

- **Goal:** Convert problem into LP
  - Approximate convex function by $K$ (PWL) segments
  - Epigraph minimization ($p$ variables)

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{L} p_i \\
\text{subject to} & \quad Ax^{(d)} = s^{(d)} \\
 & \quad x^{(d)} \geq 0 \\
 & \quad -p_i + m_i^k T_i \leq -b_i^k, \quad \forall i = 1, \ldots, L, \forall k = 1, \ldots, K \\
 & \quad T \leq C,
\end{align*}
\]

where $d = \{1, \ldots, D\}$,
$m_i^k$ is the slope and $b_i^k$ is the intercept of the $k$th PWL approximation on the link $i$. 

- $[\text{Image} 36x36$ to $46x46]$
- $[\text{Image} 306x92$ to $538x700]$
Cost Function

- Minimize delay over all links
- Delay for one link
  - M/M/1 queue delay
    \[ \phi_i(T_i) = \frac{T_i}{C_i - T_i} \]
  - A convex function
- Problem
  - Complex algorithms
  - Slow convergence
PWL Approximation

- How to approximate?
  - Uniform
  - MSE
  - Min-Max
PWL Optimization Algorithm

- Centralized “one-shot” algorithm ($K > 100$)
  - Computational intensive
  - Very accurate results for underutilized networks

- Centralized iterative algorithm ($K < 10$)
  - Solve LP for the given $K$ (start)
  - Identify link segment $k = 1, ..., K*$ that contains traffic flow
  - Split marked link segment into $K$ more segments
  - Update LP constraints (slopes and intercepts)
  - Repeat until stopping criteria satisfied
Algorithm Simulation

- Experimental Setup
  - MATLAB `linprog()`
  - Sprint IP backbone network topology
  - Traffic matrix
    - Uniform traffic
    - Sparsity pattern
Results

- Uniform traffic, unit capacities, \((K = 2, 3, 5)\)
Results

- Iterations (how many?)
  - Stopping criteria
- Feasability
  - 10E-6
- Convergence
  - Always finds feasible solution (if one exists)
More Results

- Gaussian traffic with sparsity pattern, unit C
Even More Results

- Heavy Gaussian traffic with sparsity, random C
Traffic Distribution

Total flow allocation

[Diagram showing network with nodes labeled 1 to 15 and edges connecting them, illustrating traffic distribution.]
Computational Complexity

- LP interior-point algorithms
  - $O(M^{3.5}D)$ number of arithmetic operations
  - $O(\sqrt{MD})$ number of iterations
  - $M$ is number of variables + inequality constraints
- For our problem:
  - $M = LF + LK + L$
  - For $i$ iterations = $iL(K+F+i/2+3/2)$
Storage Complexity

- Memory storage requirements
  - $\left( LFN + N + LF \right) (\lfloor \log_2 D \rfloor + 2)$
    - $F$ – Flows
    - $L$ – Links
    - $N$ – Nodes
    - $K$ – number of segments
  - $\alpha = \lfloor \log_2 D \rfloor + 2$
  - $(KLFN + N + KLF) \alpha \approx KLF(N + 1)\alpha$
  - $i$th iteration: $(K + i)LF(N + 1)\alpha$
Distributed algorithm

- Centralized implementation
- Easily distributed (especially PWL approximation)
- Dual methods
  - Subgradient ascent
  - Lagrangian relaxation
- Path augmentation approach
Protocol Implementation

- Routing protocols
  - MPLS-like labeling
  - At most $M$ flows
  - $M$ – Number of edge routers or PoPs
  - DEST determines $p_i$

<table>
<thead>
<tr>
<th>DEST</th>
<th>IP PACKET</th>
</tr>
</thead>
</table>

$p_1$  
$p_2$  
$p_3$
Edge Routers

- Full look-up
  - DEST – ID of edge router where packet leaves PoP
  - Identifies flows within PoP

- Congestion Control
  - All congestion control can be done at the edges
  - Detect when traffic not admissible – not feasible
    - Drop packets at edge
  - Estimate flows – recalculation at substantial change
  - Should not occur often – large aggregation of flows
Conclusion

- Most links underutilized (IP backbone)
- But still cannot guarantee performance (e.g. delay)
- Optimal network flow allocation could help
- We suggest a practical algorithm
  - Converges to near optimal solutions
  - Few iterations
  - Standard LP
  - Can make it distributed
- Special thanks to Yashar Ganjali for all his help!!
Questions?
PWL Approximation (1)

- Convert PWL problem into LP

new objective with PWL approximations

\[
\text{minimize } \sum_{i=1}^{L} \max_{k=1,\ldots,K} \left( m_i^k T_i + b_i^k \right).
\]

- write each of these nonlinear constraints as

\[
K \text{ linear constraints in the form } m_i^k T_i + b_i^k \leq p_i.
\]

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{L} p_i \\
\text{subject to} & \quad Ax^{(d)} = s^{(d)} \\
& \quad x^{(d)} \geq 0 \\
& \quad -p_i + m_i^k T_i \leq -b_i^k, \quad \forall i = 1, \ldots, L, \quad \forall k = 1, \ldots, K \\
& \quad T \leq C,
\end{align*}
\]

where \( d = \{1, \ldots, D\}. \)