Overlay networks maximizing throughput

Olivier Beaumont, Lionel Eyraud-Dubois, Shailesh Kumar Agrawal

Cepage team, LaBRI, Bordeaux, France

IPDPS
April 20, 2010
Outline

1. Introduction
2. Complexity
3. Successive algorithms
4. Simulations
5. Conclusions
In this talk: broadcast/streaming operation

- One source node holds (or generates) a message
- All nodes must receive the complete message
- Steady-state: quantity of data per time unit
- Goal: optimize throughput
Communication model

Explore the Bounded Multi Port model

- P2P setting, Application-Level: no \textit{a priori} communication network
- Simultaneous communications, with a per-node bandwidth bound
- Internet-like: no contention inside the network
- Steady-state approach

- Goal of algorithms: build an (efficient) overlay
- Keep things reasonable: degree constraint
Explore the Bounded Multi Port model

- P2P setting, Application-Level: no \textit{a priori} communication network
- Simultaneous communications, with a per-node bandwidth bound
- Internet-like: no contention inside the network
- Steady-state approach

- Goal of algorithms: build an (efficient) overlay
- Keep things reasonable: degree constraint
An example

\[ b_0 = 2 \]

\[ N_0 \]

\[ b_1 = 1 \]

\[ N_1 \]

\[ b_2 = 1 \]

\[ N_2 \]
An example

Best tree: $T = 1$
Best DAG: $T = 1.5$
Optimal: $T = 2$
Precise model

An instance
- $n$ nodes, with output bandwidth $b_i$ and maximal out-degree $d_i$
- node $N_0$ is the master node that holds the data

A solution (Trees)
- A weighted set of spanning trees $(w_k, T_k)$
- $\forall j, \sum_k \sum_i \chi_k(N_j, N_i)w_k \leq b_j$ (capacity constraint at node $j$)
- $\forall j, \sum_i \max_k \chi_k(N_j, N_i) \leq d_j$ (degree constraint at node $j$)
- Maximize $T = \sum_k w_k$
Precise model

An instance
- $n$ nodes, with output bandwidth $b_i$ and maximal out-degree $d_i$
- node $N_0$ is the master node that holds the data

A solution (Flows)
- Flow $f_{j}^{i}$ from node $N_j$ to $N_i$
- $\forall j$, $\left|\{i, f_{j}^{i} > 0\}\right| \leq d_j$  degree constraint at $N_j$
- $\forall j$, $\sum_i f_{j}^{i} \leq b_j$  capacity constraint at $N_j$
- Maximize $T = \min_j \text{mincut}(N_0, N_j)$
NP-Hardness

3-Partition

- 3p integers $a_i$ such that $\sum_i a_i = pT$
- Partition into $p$ sets $S_l$ such that $\sum_{i \in S_l} a_i = T$
NP-Hardness

### 3-Partition

- 3\(p\) integers \(a_i\) such that \(\sum_{i} a_i = pT\)
- Partition into \(p\) sets \(S_l\) such that \(\sum_{i \in S_l} a_i = T\)

### Reduction

- \(p\) "server" nodes, \(b_j = 2T\) and \(d_j = 4\)
- 3\(p\) "client" nodes, \(b_{j+p} = T - a_j\) and \(d_{j+p} = 1\)
- 1 "terminal" node, \(b_{4p} = 0\), \(d_{4p} = 0\)
Outline

1 Introduction

2 Complexity

3 Successive algorithms
   - Acyclic Algorithm
   - With cycles

4 Simulations

5 Conclusions
Upper bound

If $S$ has throughput $T$

- Node $N_i$ uses at most $X_i = \min(b_i, Td_i)$
- Total received rate: $nT$
- Thus $\sum_{i=0}^{n} \min(b_i, Td_i) \geq nT$
- Of course, $T \leq b_0$

Our algorithms

- Inputs: an instance, and a goal throughput $T$
- Output: a solution with resource augmentation (additional connections allowed)
Acyclic algorithm

If $\sum_{i=0}^{n-1} \min(b_i, Td_i) \geq nT$

- Order nodes by capacity: $X_1 \geq X_2 \geq \cdots \geq X_n$
- Each node $k$ sends throughput $T$ to as many nodes as possible, in consecutive order

\[ \mathcal{N}_0 \quad \mathcal{N}_1 \quad \mathcal{N}_2 \quad \mathcal{N}_3 \quad \mathcal{N}_4 \quad \mathcal{N}_5 \]
Acyclic algorithm

If $\sum_{i=0}^{n-1} \min(b_i, Td_i) \geq nT$

- Order nodes by capacity: $X_1 \geq X_2 \geq \cdots \geq X_n$
- Each node $k$ sends throughput $T$ to as many nodes as possible, in consecutive order

Ordering of nodes and throughput distribution:

$\mathcal{N}_0$ $\mathcal{N}_1$ $\mathcal{N}_2$ $\mathcal{N}_3$ $\mathcal{N}_4$ $\mathcal{N}_5$
**Acyclic algorithm**

If \( \sum_{i=0}^{n-1} \min(b_i, Td_i) \geq nT \)

- Order nodes by capacity: \( X_1 \geq X_2 \geq \cdots \geq X_n \)
- Each node \( k \) sends throughput \( T \) to as many nodes as possible, in consecutive order

![Diagram showing node order and throughput distribution](image-url)
If $\sum_{i=0}^{n-1} \min(b_i, Td_i) \geq nT$

- Order nodes by capacity: $X_1 \geq X_2 \geq \cdots \geq X_n$
- Each node $k$ sends throughput $T$ to as many nodes as possible, in consecutive order

\[ L. \ Eyraud-Dubois \ (LaBRI, \ Bordeaux) \]
**Acyclic Algorithm**

If $\sum_{i=0}^{n-1} \min(b_i, Td_i) \geq nT$

- Order nodes by capacity: $X_1 \geq X_2 \geq \cdots \geq X_n$
- Each node $k$ sends throughput $T$ to as many nodes as possible, in consecutive order

![Diagram showing the flow of nodes and throughput sending](image)
**ACYCLIC algorithm**

If \( \sum_{i=0}^{n-1} \min(b_i, Td_i) \geq nT \)

- Order nodes by capacity: \( X_1 \geq X_2 \geq \cdots \geq X_n \)
- Each node \( k \) sends throughput \( T \) to as many nodes as possible, in consecutive order

Provides a valid solution

- \( b_0 \geq T \)
- Sort by \( X_i \) \( \implies \forall k, \sum_{i=0}^{k} X_i \geq (k + 1)T \)
- Since \( X_k \leq Td_k \), the outdegree of \( N_k \) is at most \( d_k + 1 \)
Successive algorithms

With cycles

General case: \( \sum_{i=0}^{n} X_i \geq nT \)

- Acyclic algorithm until \( k_0 \) such that \( \sum_{i=0}^{k_0} X_i < (k_0 + 1)T \)
General case: \[ \sum_{i=0}^{n} X_i \geq nT \]

- Acyclic algorithm until \( k_0 \) such that \( \sum_{i=0}^{k_0} X_i < (k_0 + 1)T \)
Successive algorithms

General case: $\sum_{i=0}^{n} X_i \geq nT$

- Acyclic algorithm until $k_0$ such that $\sum_{i=0}^{k_0} X_i < (k_0 + 1)T$
Successive algorithms With cycles

General case: \[ \sum_{i=0}^{n} X_i \geq nT \]

- Acyclic algorithm until \( k_0 \) such that \( \sum_{i=0}^{k_0} X_i < (k_0 + 1)T \)
General case: \( \sum_{i=0}^{n} X_i \geq nT \)

- Acyclic algorithm until \( k_0 \) such that \( \sum_{i=0}^{k_0} X_i < (k_0 + 1)T \)
General case: \[ \sum_{i=0}^{n} X_i \geq nT \]

- Acyclic algorithm until \( k_0 \) such that \( \sum_{i=0}^{k_0} X_i < (k_0 + 1)T \)
- Recursively build partial solutions in which
  - All nodes up to \( N_k \) are served
  - Only node \( N_k \) has remaining bandwidth

Notations:

- \( M_k = kT - \sum_{i=0}^{k-1} X_i \)
- \( M_k + R_k = X_k \)
- \( M_{k+1} = T - R_k \)

L. Eyraud-Dubois (LaBRI, Bordeaux)
Recursion: initial case

Set of nodes

\[ T - M_{k_0} \]

\[ f^v_u \]

Set of nodes

\[ k_0 \]

\[ k_0 + 1 \]

Remind

1. \[ M_k + R_k = X_k \]
2. \[ R_k + M_{k+1} = T \]
3. \[ f^v_u \geq M_{k_0} \]
Recursion: initial case

Set of nodes

\[ k_0 \]
\[ k_0 + 1 \]

\[ f_{u} - M_{k_0} \]
\[ f_{u} \geq M_{k_0} \]
\[ R_{k_0} + \beta \]
\[ \beta = \frac{M_{k_0+1}}{T} M_{k_0} \]
\[ \alpha = \frac{M_{k_0+1}}{T} \left( T - M_{k_0} \right) \]
\[ f_{k_0+1} + f_{k_0} ^{k_0+1} = T \]

\[ T - M_{k_0} - \alpha \]
\[ M_{k_0} - \beta \]

\[ \alpha \]
\[ \beta \]

\[ u \]

\[ v \]

\[ M_{k} + R_{k} = X_{k} \]
\[ R_{k} + M_{k+1} = T \]

Remind
Recursion: general case

\[ f_k(k-1) = T - f_{k-1} \]

\[ f_k(k) = T - f_k(k-1) \]

\[ \alpha = M_{k+1} + f_k(k) \]

\[ \beta = M_{k+1} + f_{k-1}^k \]

\[ \alpha + \beta = M_{k+1} + f_k(k) + f_{k-1}^k = T \]

\[ R_k + M_{k+1} = T \]

\[ M_k + R_k = X_k \]
Recursion: general case

\[
\begin{align*}
T - f_{k-1}^k & = f_k^k - \alpha \\
T - f_{k-1}^k & = R_k + \beta \\
R_{k+1} & = f_{k-1}^k + f_k^k
\end{align*}
\]

Remind

\[
\begin{align*}
M_k + R_k & = X_k \\
R_k + M_{k+1} & = T
\end{align*}
\]

\[
\begin{align*}
f_{k-1}^k + f_k^k & = T \\
\beta & = \frac{M_{k+1}}{T} f_k^k \\
\alpha & = \frac{M_{k+1}}{T} f_{k-1}^k \\
\alpha + \beta & = M_{k+1} \\
f_{k+1}^k + f_k^{k+1} & = T
\end{align*}
\]
Recursion: result

**Final outdegree of** $\mathcal{N}_i$: $o_i \leq \max(d_i + 2, 4)$

- Acyclic solution: $o_i \leq d_i + 1$
- Degree of $\mathcal{N}_u$ and of one node in the set is increased by 1
- Step $k$: $o_k$ and $o_{k-1}$ are increased by 1
Outline

1. Introduction
2. Complexity
3. Successive algorithms
4. Simulations
5. Conclusions
Comparison of different solutions

Unconstrained solution

Best achievable throughput without degree constraints: \( \frac{\sum_i b_i}{n} \)

Best Tree

In a tree of throughput \( T \), flow through all edges must be \( T \). Counting the edges yield \( \sum_i \min(d_i, \left\lfloor \frac{b_i}{T} \right\rfloor) \geq n \).

Best Acyclic

Computed by the ACYCLIC algorithm

Cyclic

Throughput when adding cycles
Simulations

Results: comparisons to Cyclic

Overlay networks maximizing throughput 18/21
Results: Cyclic vs Unconstrained

Cycle ratio against Optimal

Throughput ratio vs Output degree for different values of N (N=10, N=100, N=1000).
Summary

- Theoretical study of the problem: optimal resource augmentation algorithm
- In practice:
  - a low degree is enough to reach a high throughput
  - an acyclic solution is very reasonable
  - once the overlay is computed, there exist distributed algorithms to perform the broadcast

Going further

- Worst-case approximation ratio of ACYCLIC?
- Study the robustness of our algorithms
- Design on-line and/or distributed versions