

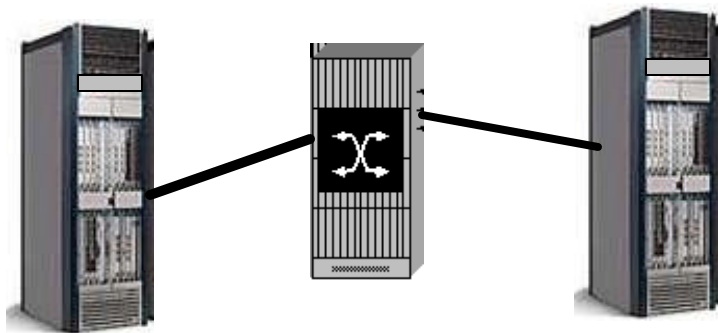
Efficient Multicast on a Terabit Router

Punit Bhargava punit@cisco.com

Sriram C. Krishnan srikrish@cisco.com

Rina Panigrahy rinap@cisco.com

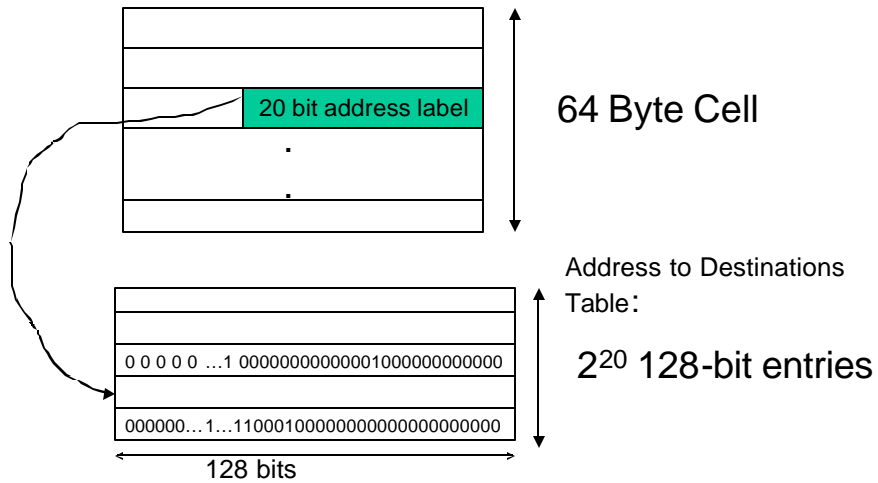
Distributed Router



128 linecard: 5 Terabit system

Need 7 bits to specify address unicast

128 bits (16 bytes) to address multicast (exactly)



Need to supercast efficiently

Number of multicast connections



Run out of space in address-to-destinations table

Distinct destination sets have to "share" an entry in table: program the *super* set (union)

1011..... } 1111.....
0110..... }

Supercast: send to more linecards and discard at non-subscribing linecards

Talk Outline

Introduction/Motivation

Problem Formulation

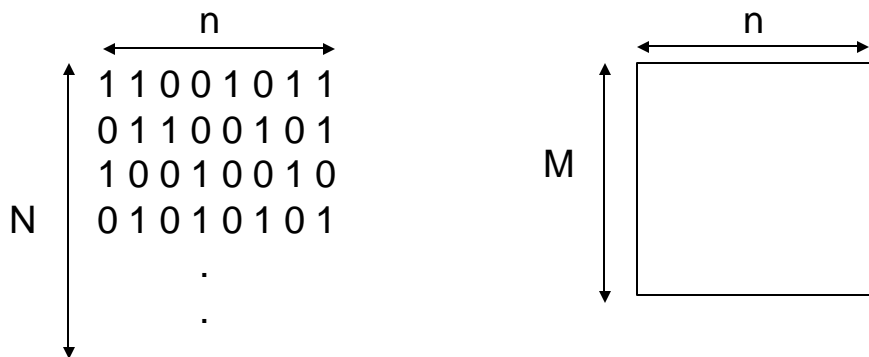
Computational Complexity

Heuristic Algorithms

Experimental Results

Conclusions

Minimum Cumulative Supercast (MCS) Problem



Given N destinations sets (DS), positive integers K , M , such that $M < N$,

does there exist an M -clustering (assignment of each DS to a cluster, $f: N \rightarrow M$)

such that “total supercast amongst all clusters” $< K$?

Example: N=4, M=2

Amount of supercast: Number of dominated zeros

1 1 0 0 1 0 1 1	1 1 1 0 1 1 1 1	
0 1 1 0 0 1 0 1	1 1 0 0 1 0 1 1	9
1 0 0 1 0 0 1 0	0 1 1 0 0 1 0 1	
0 1 0 1 0 1 0 1	1 0 0 1 0 0 1 0	
	0 1 0 1 0 1 0 1	
	1 1 0 1 0 1 1 1	

Assigned set:

Destination set assigned for
cluster: OR/union of DSs
in cluster

1 1 0 0 1 0 1 1	12
0 1 1 0 0 1 0 1	
1 0 0 1 0 0 1 0	
0 1 0 1 0 1 0 1	

One cluster cost & cost savings of a cluster

1 1 0 0 1 0 1 1
 0 1 1 0 0 1 0 1
 1 0 0 1 0 0 1 0
 0 1 0 1 0 1 0 1

One cluster cost : total number of zeros in all
destination sets (otherwise eliminate column
of zeros)

Additional (more than one) clusters should yield
cost savings

One cluster cost & cost savings of a cluster

```
1 1 0 0 1 0 1 1
0 1 1 0 0 1 0 1
1 0 0 1 0 0 1 0
0 1 0 1 0 1 0 1
```

No cost savings

One cluster cost : total number of zeros in all destination sets

One cluster cost & cost savings of a cluster

```
1 1 0 0 1 0 1 1
0 1 1 0 0 1 0 1
1 0 0 1 0 0 1 0
0 1 0 1 0 1 0 1
```

One cluster cost : number of zeros in Destinations Matrix (otherwise eliminate column of zeros)

Cluster with column of zeros yields savings

Cost savings = columns of zeros

Talk Outline

Introduction/Motivation

Problem Formulation

Computational Complexity

Heuristic Algorithms

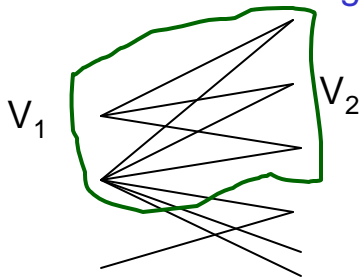
Experimental Results

Conclusions

Minimum Cumulative Supercast is NP-Complete

Maximum Edge Biclique [Peeters 2003]:

Given a bipartite graph, find complete subgraph with maximum edge count

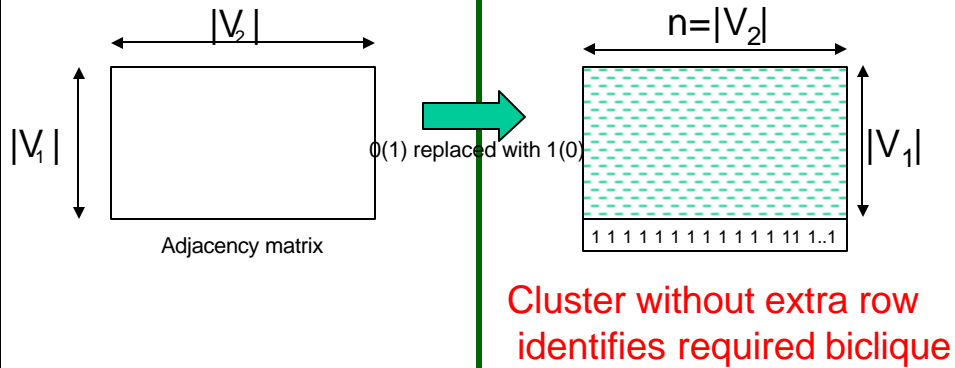


Complete (sub)-
graph has all ones
adjacency matrix

Reduction for NP-hardness of MCS

Given bipartite graph
 $G = (V_1, V_2, E \subseteq (V_1 \times V_2))$,
 positive integer K' ;
 Is there a biclique with $\geq K'$
 Edges?

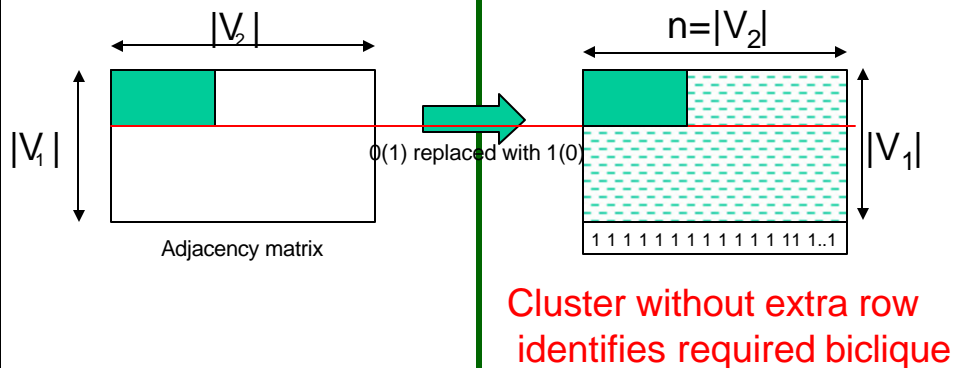
$N = |V_1| + 1$
 $n = |V_2|$
 $M = 2$ (want two clustering)
 $K = |E| + 1 - K'$



Yes-instance of MEB ==> Yes-instance of MCS

Given bipartite graph
 $G = (V_1, V_2, E \subseteq (V_1 \times V_2))$,
 positive integer K' ;
 Is there a biclique with $\geq K'$
 Edges?

$N = |V_1| + 1$
 $n = |V_2|$
 $M = 2$ (want two clustering)
 $K = |E| + 1 - K'$



Hardness of Approximation

Theorem: It is RSAT-hard to approximate MCS to a factor better than $31/28$.

Talk Outline

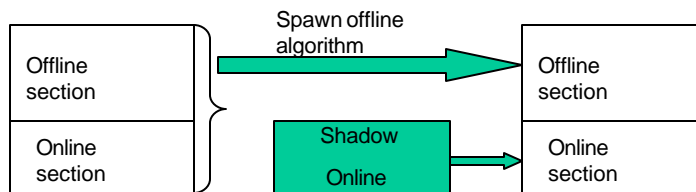
Introduction/Motivation
Problem Formulation
Computational Complexity
Heuristic Algorithms
Experimental Results
Conclusions

Algorithm types: online and offline

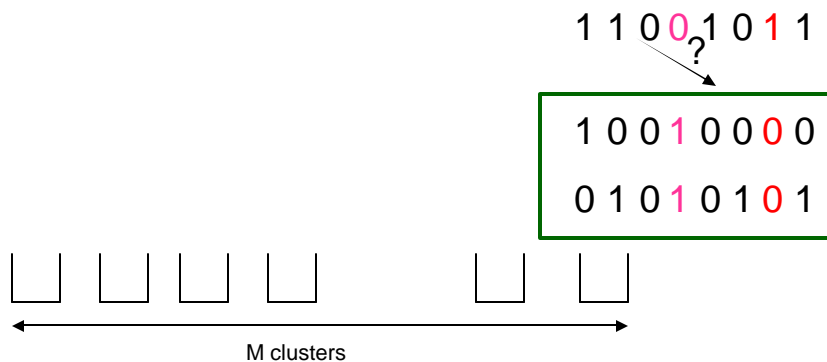
Online algorithm is necessary

Switch Fabric provides extra copy of address to destinations table:

Split up label to destinations table: online and offline section



Online greedy row clustering

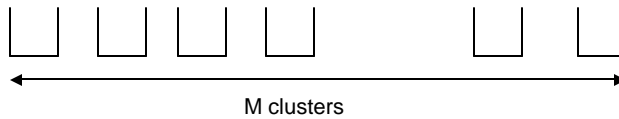


Pick the cluster that results in least cost increase

Two components to cost increase

$O(Mn)$ time to decide cluster choice

Offline greedy row clustering: two greedy alg.



Can we discover well clustered inputs?

Random seeding of M clusters will miss $1/e$ clusters

Choose target of $4M$ clusters:

Distribute N destination sets among $4M$ clusters

Greedy pick best pair-wise merge to reduce from $4M$ to M clusters

$O(N.M.n)$

Column clustering

1	1	0	0	1	0	1	1
0	1	1	0	0	1	0	1
1	0	0	1	0	0	1	0
0	1	0	1	0	1	0	1

Tradeoff quality for run-time

Split columns in two clusters: $n/2$ columns in each

Target number of clusters per instance: $M^{1/2}$

Online complexity: $O(M^{1/2}.n)$

Offline complexity: $O(N.M^{1/2}.n)$

constant time algorithms

$m = \log M$; number of bits to represent cluster number

$O(m)$: practically constant time

Compute m bit signature (cluster number)

Signature: a proximity measure

Destination sets that don't differ by much
will have same signature

Subset Intersection Signature (SIS)

00100000

11001011

01100101

10010010

01010101

Pick m random subsets of $\{1, 2, \dots, n\}$

Each of m bits of signature is determined based
on intersection with corresponding subset

Choice of subset can exploit knowledge of input
distribution

Column classification: special case of SIS

1 1 0 0 1 0 1 1

0 1 1 0 0 1 0 1

1 0 0 1 0 0 1 0

0 1 0 1 0 1 0 1

If each of the subsets is distinct and a singleton,
amounts to picking m columns and classifying
based on these

Easy method to cluster

Random Permutation Signature (RPS)

Pick m random permutations of $\{1, 2, \dots, n\}$

Subject the DS to the m permutations

Let i_1, i_2, \dots, i_m be the index of the
first zero (or one) location in the permuted DS

Form m bit signature:

Can choose more bits from one index and use
fewer indices or fewer bits from each and more
indices

Example: if $n=256$ and $m=10$; Each min-index is
8 bits wide.

Which algorithm to choose?

Choose the best algorithm (online) the connection rate can afford

Resort to more expensive algorithm offline

Talk Outline

Introduction/Motivation

Problem Formulation

Computational Complexity

Heuristic Algorithms

Experimental Results

Conclusions

Classes of inputs

1. Universe of inputs: $N = 2^n$
2. Bernoulli IID inputs: each of n destinations in a destination set turned on independently with probability p
3. Pre-clustered inputs

What we measure

Average Bandwidth Waste: $\text{total-cost}/Nn$

What fraction of linecards (n) receive supercast, i.e., packets to drop

One-cluster average bandwidth waste is fraction of zeros in a destination set:

Bernoulli IID inputs: $1-p$

Universe of inputs: 0.5

Universe of inputs

For the universe of inputs, on average a destination set has $n/2$ zeros

Column classification by m columns achieves:

Average BW waste: $(n-m)/2n$

Average BW savings : $m/2n$

Theorem: No 2^m -clustering for universe of inputs can achieve average BW savings greater than m/n .

Universe of Inputs

n	m	greedy	Column-classification : $(n-m)/2n$	RPS	2-Column Clusters	SIS
14	4	0.342	0.357	0.464	0.353	0.426
16	8	0.206	0.250	0.453	0.230	0.334
20	10	0.200	0.250	0.456	0.224	0.372

SIS: Pick subsets with $1/p$ (here $p = 1/2$) ones

Random (Bernoulli IID) inputs

Theorem: Given N destination sets over n destinations with each destination chosen independently and with probability p and $M (= 2^m)$ possible clusters, then:

Any clustering can achieve no more cost savings over the one cluster cost than $O(1/p(N.\log M + Mn))$

There is a clustering which achieves a cost savings of $\Omega((1-p)(N.\log M + Mn))$

Random inputs: bounds on BW savings

Total BW savings

Average BW savings

$$O(1/p(N.\log M + Mn))$$

$$O(1/p(\log M/n + M/N))$$

$$\Omega((1-p)(N.\log M + Mn))$$

$$\Omega((1-p)(\log M/n + M/N))$$

When p is a constant close to 0 or 1 (e.g., 0.5): bounds match (upto constant factors)

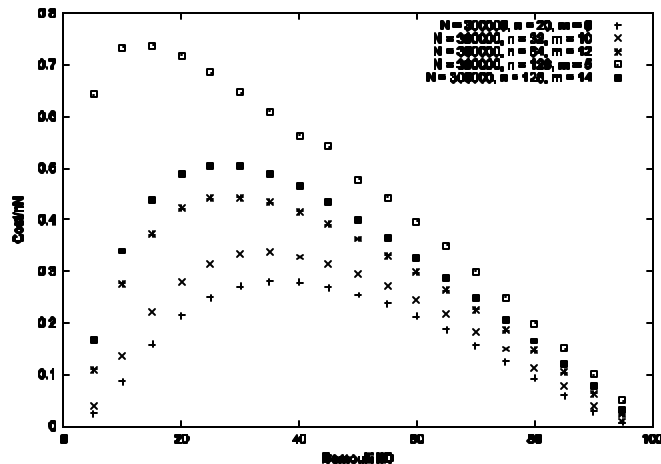
When number of destination-sets per cluster (N/M) is high, column-classification (first term) dominates

Random Inputs: Table of results (Avg BW waste)

Prob. p	1 - p	greedy y	2 column clusters	RPS	SIS	Col. Classif:
0.1	0.9	0.241	0.325	0.846	0.441	0.713
0.3	0.7	0.427	0.471	0.688	0.563	0.569
0.5	0.5	0.354	0.379	0.499	0.427	0.406
0.7	0.3	0.220	0.247	0.300	0.269	0.244
0.9	0.1	0.061	0.070	0.100	0.095	0.081

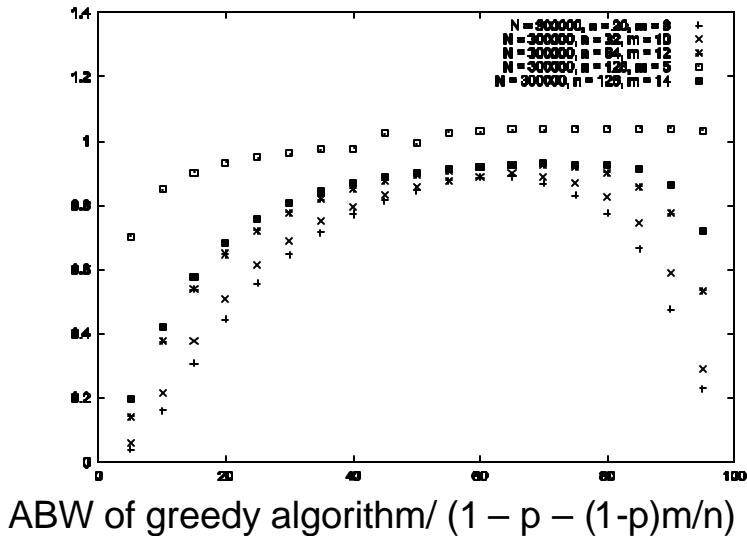
N=100,000 (DSs); n=64 (linecards); m=12 (4096 clusters)
 Column classification: $1 - p - (1-p)m/n$

Random inputs: Plot of Avg BW waste with greedy algorithm vs prob p



N/M at least 300,000/16000;
 savings proportional to m/n (waste to n/m)

Random Inputs: plot of ratio of avg BW waste of greedy and col. Classific. Vs probability p



Preclustered inputs: comparison of Avg BW waste from different methods

p_1 %	p_2 %	Expt ABW	Grdy M-cl	G-4M	G- 4M- M	2 Col Cluster s	RPS	SIS	1- $p_1 * p_2$
30	30	0.206	0.395	0.232	0.273	0.546	0.828	0.615	0.91
30	70	0.088	0.171	0.082	0.085	0.601	0.743	0.656	0.79
50	50	0.248	0.433	0.248	0.251	0.604	0.711	0.655	0.75
50	70	0.149	0.262	0.144	0.145	0.556	0.627	0.603	0.65
70	70	0.210	0.329	0.207	0.208	0.451	0.493	0.488	0.51

N=100,000 (Destination sets); n=64 (linecards) ;
m=6 (64 clusters)

Summary

Cluster DSs to fit into address to destinations table in switch fabric to minimize supercast

Proved hard to solve exactly and approximately

Proposed algorithms of varying run-time/quality:

greedy row clustering

column clustering

constant time methods: RPS & SIS (column classification)

Derived “tight” bounds for savings for random inputs

Previous Work

Marsan, Chiussi, Francini, Galante, and Leonardi,
“Compression of Multicast Labels in Large IP Routers,” *IEEE journal on selected areas in communications*, vol 21, no 4, pp 630-641, 2003.

Extended Version of paper:

With proof of tight bound on bandwidth savings for random inputs &

Greedy reduction of $4M$ to M clusters to discover pre-clustered inputs, etc,

Email authors or visit URL in paper