VirtuCast: Multicast and Aggregation with In-Network Processing

An Exact Single-Commodity Algorithm

Matthias Rost & Stefan Schmid

TU Berlin & Telekom Innovation Laboratories (T-Labs)

December 17th, 2013

Our Work in a Nutshell

Virtualization on the rise: SDN + NFV

- How to compute virtual aggregation / multicasting trees?
- Where to place in-network processing functionality?

Our Answer

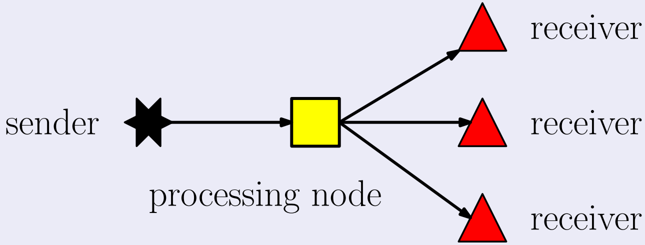
- *New Model*: Constrained Virtual Steiner Arborescence Problem
- *New Algorithm*: VirtuCast

Objective: Jointly minimize ...

- bandwidth
- number of processing nodes

Communication Schemes: Multicast

processing = duplication + reroute



Communication Schemes: Multicast

processing = duplication + reroute

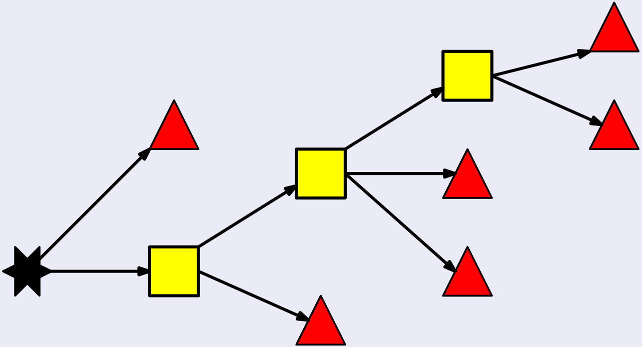
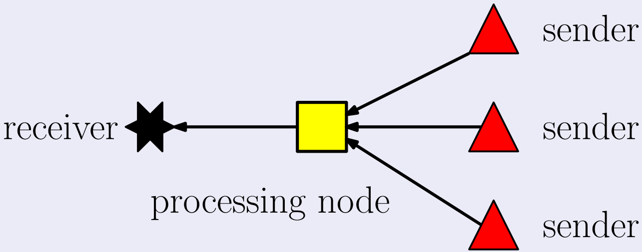


Figure: Hierarchy of processing nodes

Communication Schemes: Aggregation

processing = merge + reroute



Communication Schemes: Aggregation

processing = merge + reroute

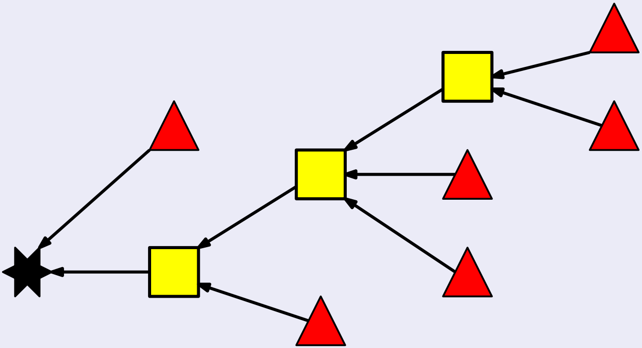


Figure: Hierarchy of processing nodes

Introductory Example

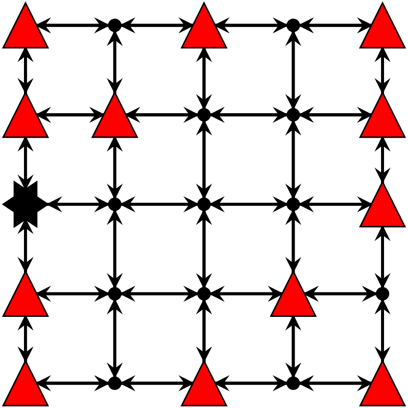
Aggregation scenario

grid graph with 14 senders and one receiver

Virtualized links

Flow can be routed along arbitrary paths

receiver | sender



Without in-network processing: Unicast

Solution Method

- minimal cost flow

Solution uses

- 43 edges
- 0 processing nodes

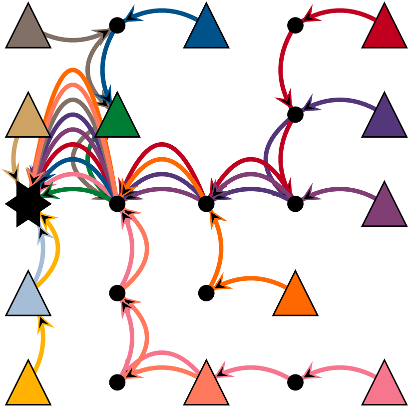


Figure: Unicast solution

With in-network processing at all nodes

Solution Method

- Steiner arborescence

Solution uses

- 16 edges
- 9 processing nodes

✶	receiver	▲	sender
■	processing node	⬠	sender with processing

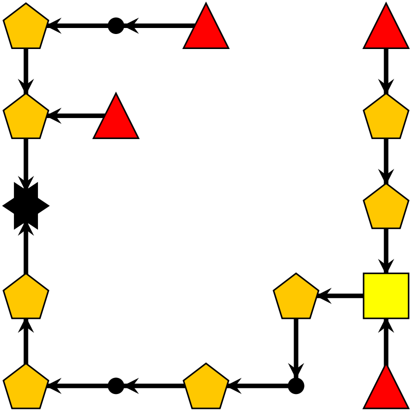
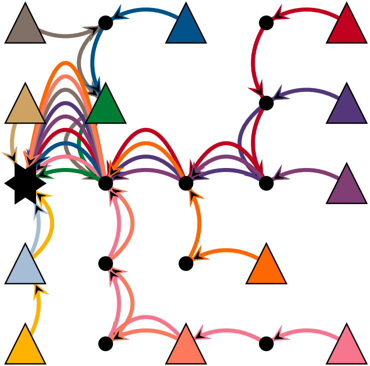
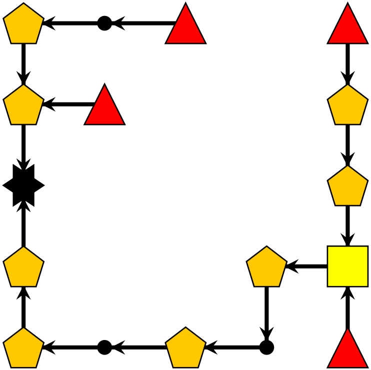


Figure: Aggregation tree

How to Trade-off?



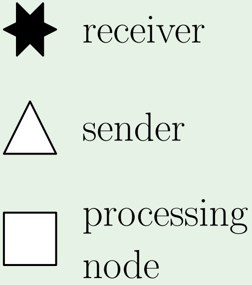
vs.

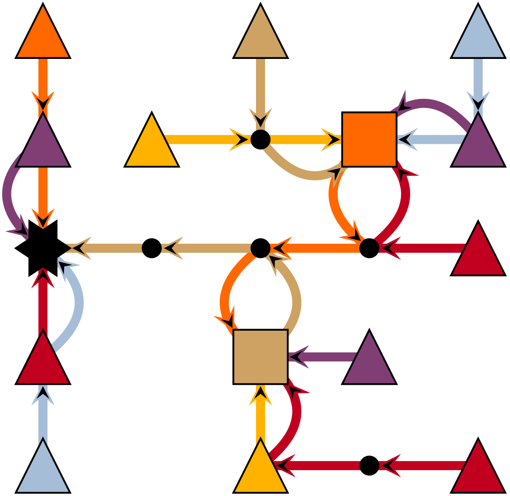


Our Solution: CVSAP & VirtuCast

Solution uses

- 26 edges
- 2 processing nodes





Our Solution: CVSAP & VirtuCast

Solution uses

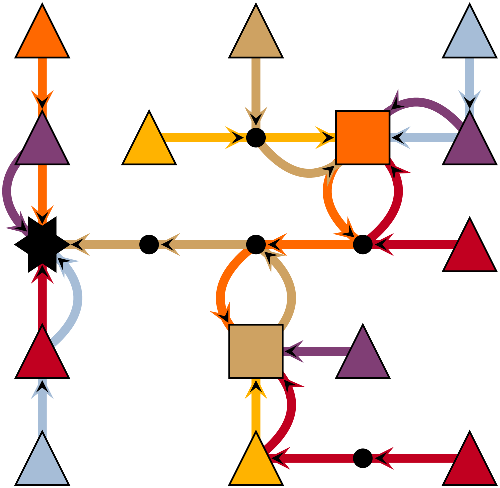
- 26 edges
- 2 processing nodes

New Model

Constrained Virtual Steiner Arboresence Problem (CVSAP)

New Solution Method

VirtuCast algorithm



Definition of the
Constrained Virtual Steiner Arborescence Problem

Multicast $\triangleq$ Aggregation

Multicasting scenario can be reduced onto the aggregation scenario

We only consider the aggregation scenario.

Input to the Constained Virtual Steiner Arborescence Problem

Graph

- Directed Graph $G = (V_G, E_G)$
- Root $r \in V_G$, i.e. single receiver
- Terminals $T \subset V_G$, i.e. sender
- Steiner sites $S \subset V_G$, i.e. potential processing locations

Input to the Constained Virtual Steiner Arborescence Problem

Graph

- Directed Graph $G = (V_G, E_G)$
- Root $r \in V_G$, i.e. single receiver
- Terminals $T \subset V_G$, i.e. sender
- Steiner sites $S \subset V_G$, i.e. potential processing locations

Important

No processing functionality can be placed on non-Steiner nodes.

Input to the Constained Virtual Steiner Arborescence Problem

Graph

- Directed Graph $G = (V_G, E_G)$
- Root $r \in V_G$, i.e. single receiver
- Terminals $T \subset V_G$, i.e. sender
- Steiner sites $S \subset V_G$, i.e. potential processing locations

Important

No processing functionality can be placed on non-Steiner nodes.

Costs

- for edges c_E
- for opening Steiner sites c_S

Capacities

- for edges u_E
- for Steiner sites & the root u_S, u_r

CVSAP Solution

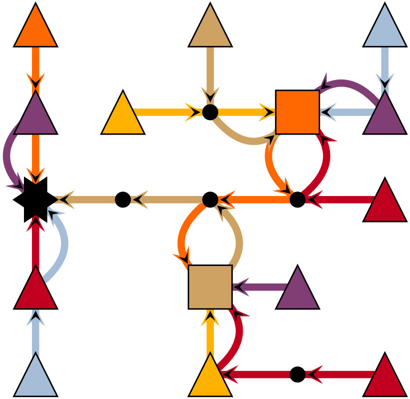
Virtual Links

sender & processing nodes are connected via *paths*

receiver

sender

processing node



Solution Structure

Virtual Arborescence

- directed tree towards root r
- terminals are leaves
- non Steiner sites are forbidden
- if a Steiner site is included, processing functionality is placed
- edges represent paths in underlying network

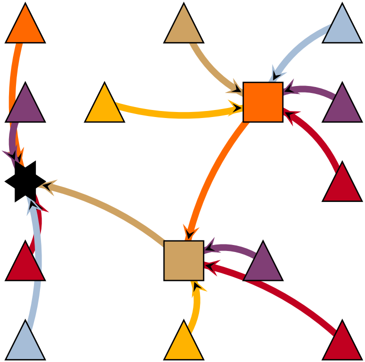


Figure: Virtual Arborescence

Constrained Virtual Steiner Arborescence Problem

Definition

Find a Virtual Arborescence such that

Degree constraints

- degrees of root r and Steiner sites are bounded by u_r and u_S

Reasoning

- aggregation nodes are not able to handle arbitrary many incoming flows
- multicasting nodes are not able to duplicate an incoming stream arbitrarily many times

Constrained Virtual Steiner Arborescence Problem

Definition

Find a Virtual Arborescence such that

- Degree constraints

Edge capacities

- edge capacities in the underlying network are not violated

Constrained Virtual Steiner Arborescence Problem

Definition

Find a Virtual Arborescence such that

- Degree constraints
- Edge capacities

minimizing

sum of edge costs + sum of installation costs

Applications

Applications

	Network	Application	Technology, e.g.
multicast	ISP	service replication / cache placement [7, 8]	middleboxes / NFV + SDN
	backbone	optical multicast [4]	ROADM[1] + SDH
	all	application-level multicast [10]	different
aggregation	sensor network	value & message aggregation [3, 5]	source routing
	ISP	network analytics [2]	middleboxes / NFV + SDN
	data center	big data / map-reduce [1]	SDN

[1] reconfigurable optical add/drop multiplexer

Solution Approach

Overview of Solution Approach

CVSAP

- novel problem
- inapproximable (if $P \neq NP$)

Goal: exact algorithm

- solves CVSAP to optimality
- non-polynomial runtime

Overview of Solution Approach

CVSAP

- novel problem
- inapproximable (if $P \neq NP$)

Goal: exact algorithm

- solves CVSAP to optimality
- non-polynomial runtime

Motivation for exact algorithms

- application dependent: allows trading-off runtime with solution quality, e.g. when designing new networks
- baseline for heuristics

Overview of Solution Approach

CVSAP
- novel problem
- inapproximable (if $P \neq NP$)

Goal: exact algorithm
- solves CVSAP to optimality
- non-polynomial runtime

Motivation for exact algorithms
- application dependent: allows trading-off runtime with solution quality, e.g. when designing new networks
- baseline for heuristics

Solution Approach: Integer Programming (IP)
- lower bounds are computed on-the-fly

Our Algorithms for CVSAP

Developed two different IP formulations

Multi-Commodity Flow based

- bad lower bounds
- cannot be used on large instances

Single-Commodity Flow based

- good lower bounds
- can be used to solve large instances
- VirtuCast

Single- vs. Multi-Commodity Flows

Single-Commodity Flow Formulation

- computes *aggregated* flow on edges independently of the origin
- does not represent virtual arborescence

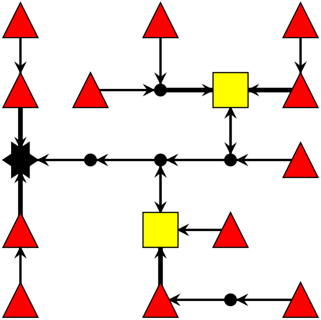


Figure: Single-commodity

Single- vs. Multi-Commodity Flows

Example: 6000 edges and 200 Steiner sites

- Single-commodity: 6000 integer variables
- Multi-commodity: 1,200,000 binary variables

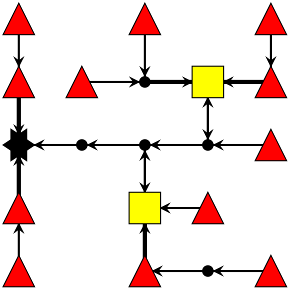


Figure: Single-commodity

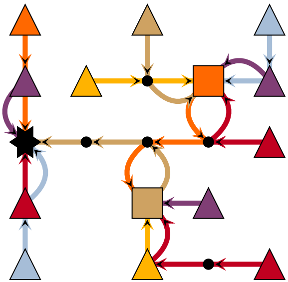


Figure: Multi-commodity

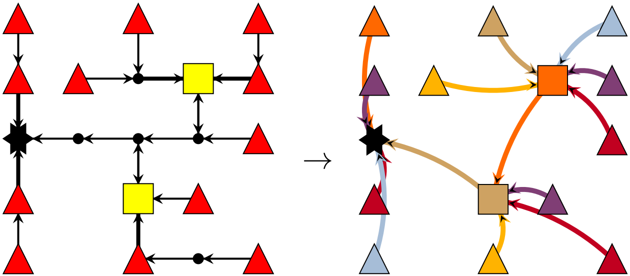
VirtuCast

VirtuCast Algorithm

Outline of VirtuCast

1. Solve single-commodity flow IP formulation.
2. Decompose IP solution into Virtual Arborescence.

How to decompose?



(a) IP solution (b) Virtual Arborescence

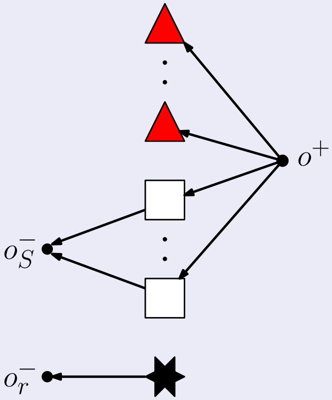
IP Formulation

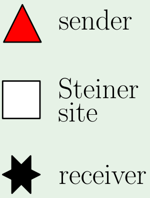
Extended Graph

Additional nodes

- source o^{+}
- sinks o_{r}^{-} and o_{S}^{-}

Additional edges





Outline of IP Formulation

Variables

$$\forall\ s \in S. \qquad x_s \in \{0,1\}$$

$$\forall\ e \in E_{\text{ext}}. \qquad f_e \in \mathbb{Z}_{\geq 0}$$

Constraints

1. single-commodity flow on extended graph
2. capacity constraints
3. connectivity inequalities

Outline of IP Formulation

Variables

$$\forall\ s \in S. \qquad x_s \in \{0,1\}$$
$$\forall\ e \in E_{\text{ext}}. \qquad f_e \in \mathbb{Z}_{\geq 0}$$

Constraints

1. single-commodity flow on extended graph
   - terminals receive one unit of flow
   - activated Steiner sites receive one unit of flow
   - flow preservation on all original nodes
2. capacity constraints
3. connectivity inequalities

Outline of IP Formulation

Variables

$$\forall\ s \in S. \qquad x_s \in \{0,1\}$$
$$\forall\ e \in E_{\text{ext}}. \qquad f_e \in \mathbb{Z}_{\geq 0}$$

Constraints

1. single-commodity flow on extended graph
2. capacity constraints
   - enforce degree constraints
   - enforce that edge capacities hold
3. connectivity inequalities

Outline of IP Formulation

Variables

$$\forall\ s \in S. \qquad x_s \in \{0,1\}$$

$$\forall\ e \in E_{\text{ext}}. \qquad f_e \in \mathbb{Z}_{\geq 0}$$

Constraints

1. single-commodity flow on extended graph
2. capacity constraints
3. connectivity inequalities

Connectivity Inequalities

$\forall\ W \subseteq V_G, s \in W \cap S \neq \emptyset.\ \ f(\delta^{+}_{E^R_{\text{ext}}}(W)) \geq x_s$

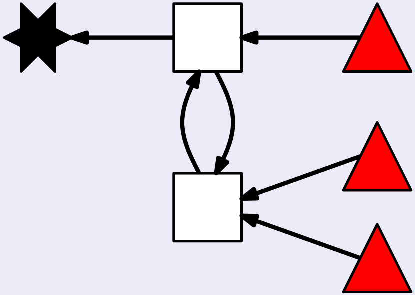
From each activated Steiner site, there exists a path towards o^-_r.

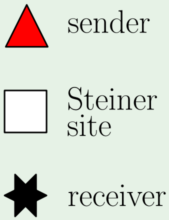
Exponentially many constraints, but . . .

can be separated in polynomial time.

Example

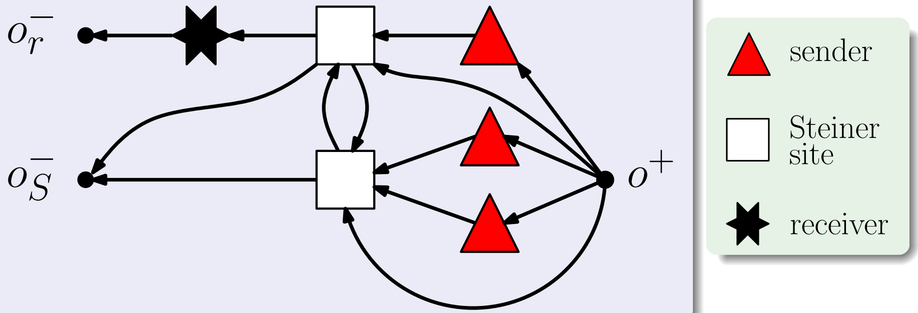
Scenario





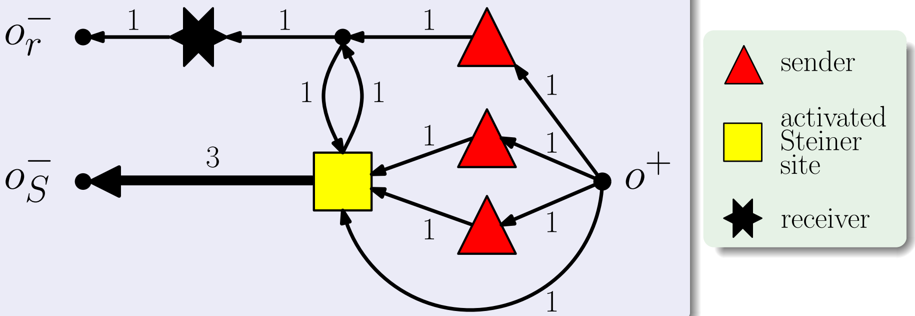
Example

Extended Graph



Example

Solution



Decomposition Algorithm

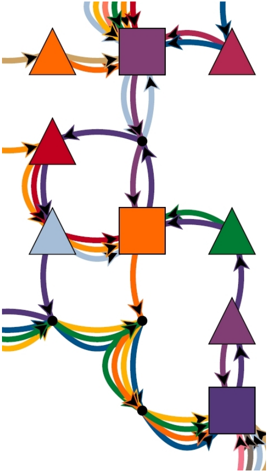
Decomposing flow is non-trivial.

Flow solution is . . .

- not a tree and
- not a DAG [6].

Flow solution . . .

- contains cycles and
- represents *arbitrary* hierarchies.



Outline of Decomposition Algorithm

Iterate

1. select a terminal t
2. construct path P from t towards o_r^- or o_S^-
3. remove one unit of flow along P
4. connect t to the second last node of P and remove t

After each iteration

Problem size reduced by one.

Outline of Decomposition Algorithm

Reduced problem must be feasible

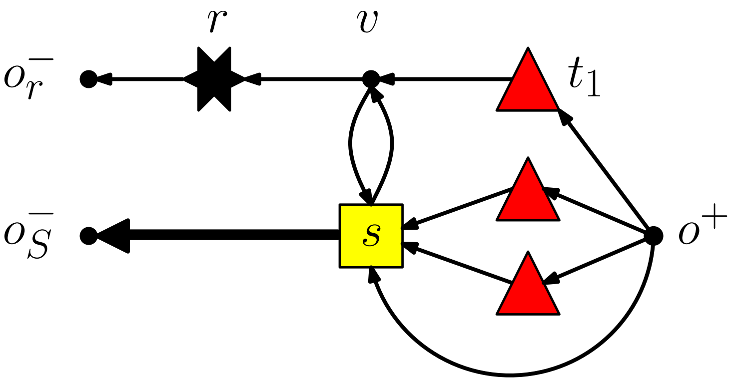
Removing flow must not invalidate any connectivity inequalities.

Principle: Repair & Redirect

- decrease flow on path edge by edge
- if connectivity inequalities are violated

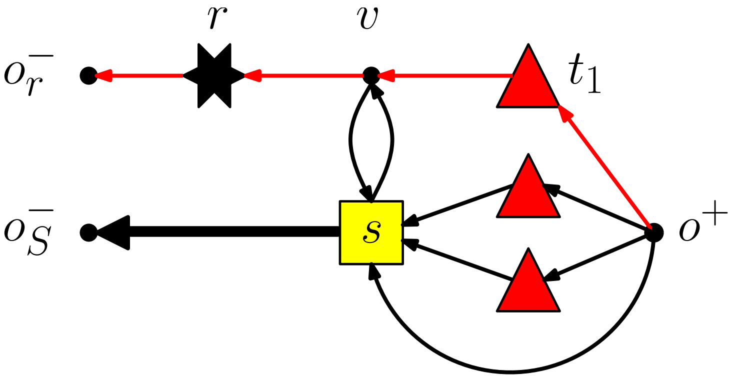
repair	increment flow on edge to remain feasible
redirect	choose another path from the current node

Decomposition Example I



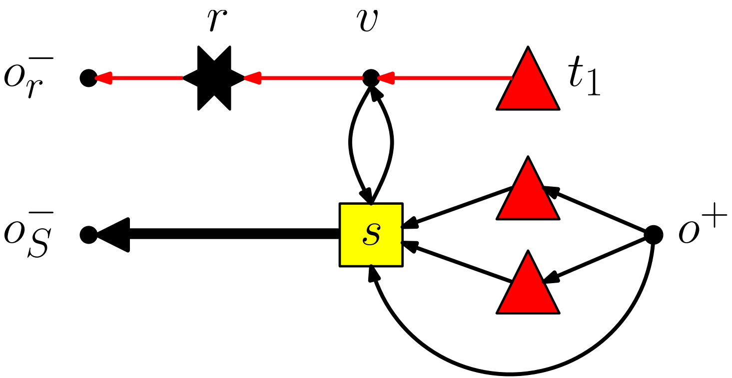
Decomposition Example I

$$P = \langle o^+, t_1, v, r, o_r^- \rangle$$



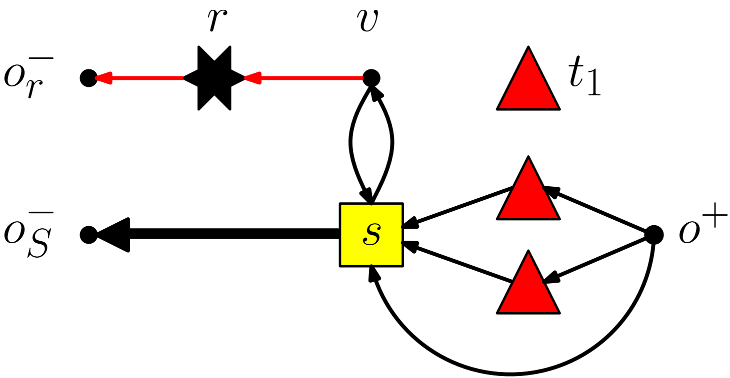
Decomposition Example I

$$P = \langle o^+, t_1, v, r, o_r^- \rangle$$



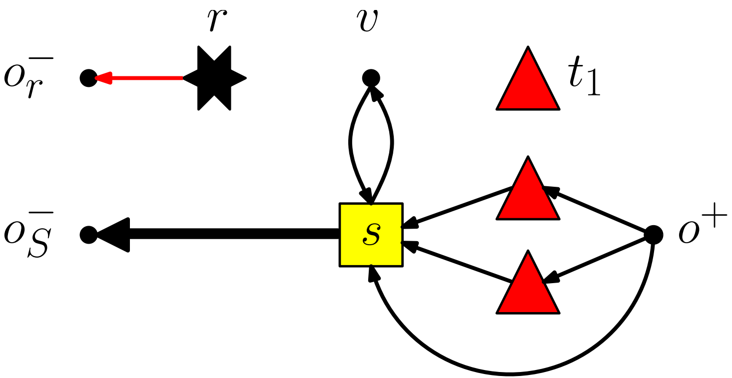
Decomposition Example I

$$P = \langle o^+, t_1, v, r, o_r^- \rangle$$

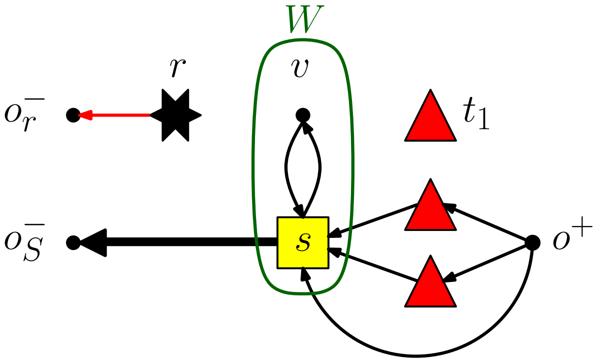


Decomposition Example I

$$P = \langle o^{+}, t_1, v, r, o_r^{-} \rangle$$



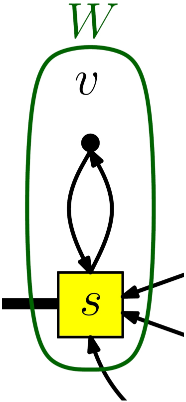
Redirecting Flow



Violation of Connectivity Inequality

$$f(\delta^+_{E^R_{\text{ext}}}(W)) \geq x_s \qquad \forall\ W \subseteq V_G, s \in W \cap S \neq \emptyset$$

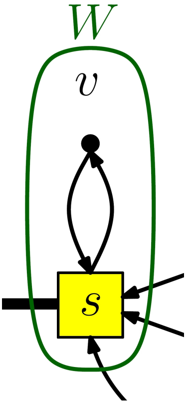
Redirecting Flow



Redirection towards o_S^- is possible!

There exists a path from v towards o_S^- in W.

Redirecting Flow



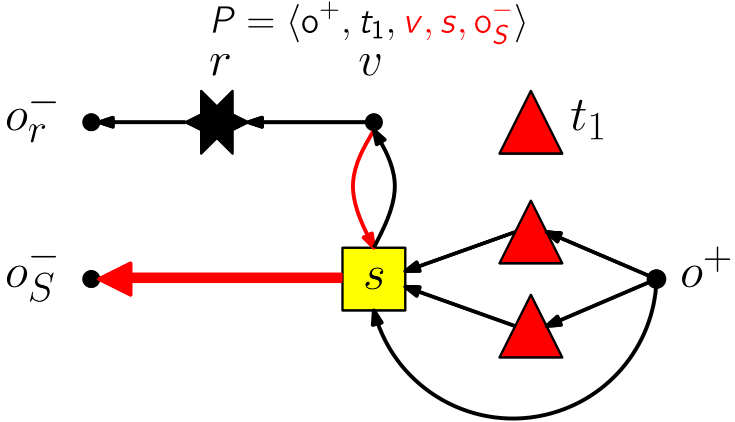
Redirection towards o_S^- is possible!

There exists a path from v towards o_S^- in W.

Reasoning

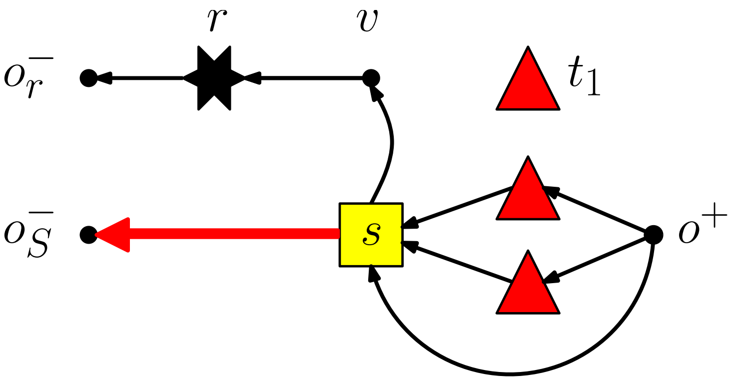
1. Flow preservation holds within W.
2. s could reach o_r^- via v before the reduction of flow.
3. v receives at least one unit of flow.
4. Flow leaving v must eventually terminate at o_S^-.

Decomposition Example II

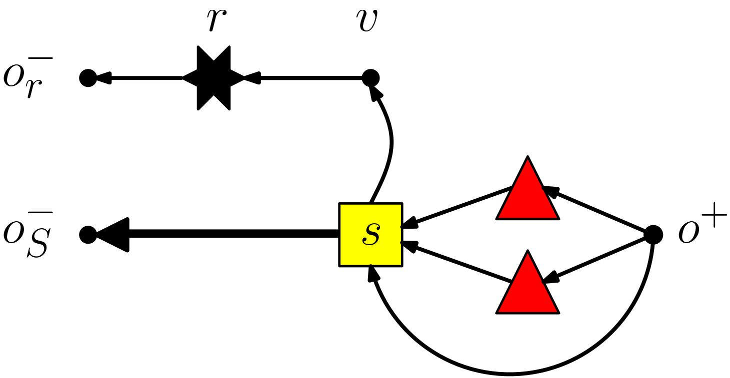


Decomposition Example II

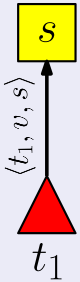
$$P = \langle o^+, t_1, v, s, o_S^- \rangle$$



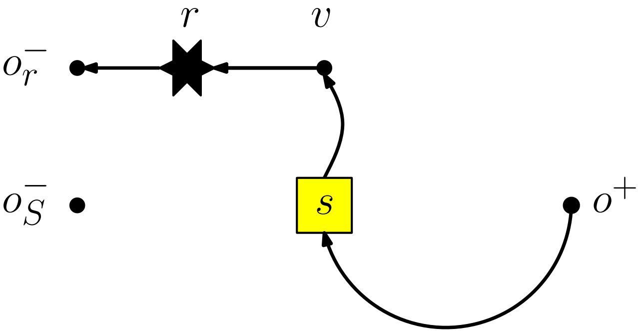
Decomposition Example II



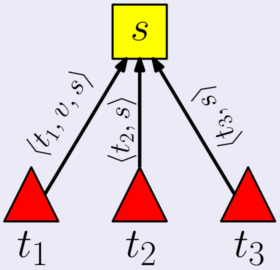
Solution



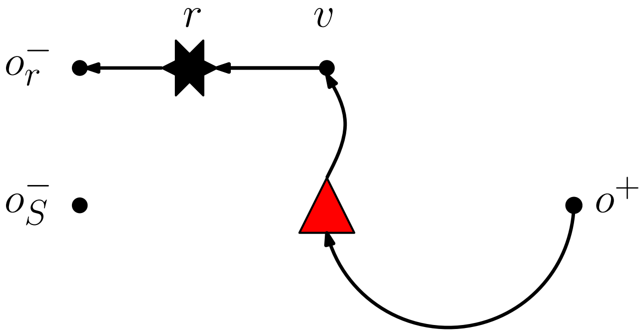
Decomposition Example II



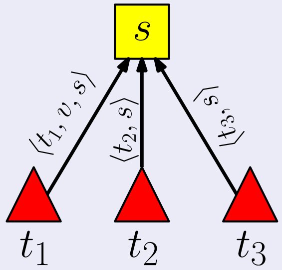
Solution



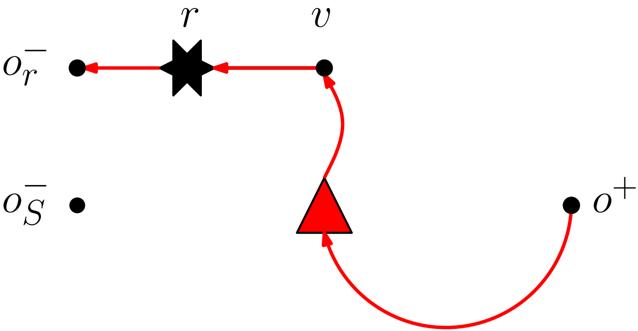
Decomposition Example II



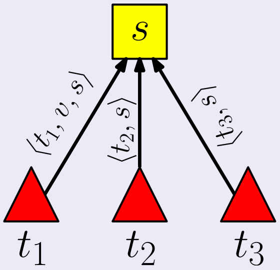
Solution



Decomposition Example II

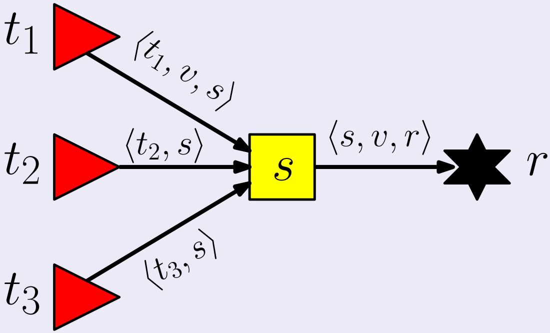


Solution



Decomposition Example II

Final Solution



Runtime of Decomposition Algorithm

Theorem

Given an optimal solution, the Decomposososition Algorithm computes a Virtual Arborescence in time $\mathcal{O}\left(|V_G|^2 \cdot |E_G| \cdot (|V_G| + |E_G|)\right)$.

Proof of Correctness

Outline of Proof

Cost-preserving mapping

$$\hat{\mathcal{T}}_G \in \mathcal{F}_{\text{CVSAP}} \underset{\text{via Decomposition algorithm}}{\overset{\text{easy}}{\rightleftarrows}} (\hat{x}, \hat{f}) \in \mathcal{F}_{\text{IP}}$$

Theorem

Algorithm VirtuCast solves CVSAP to optimality.

Computational Evaluation

Test Set: Synthetic ISP Topologies [9]

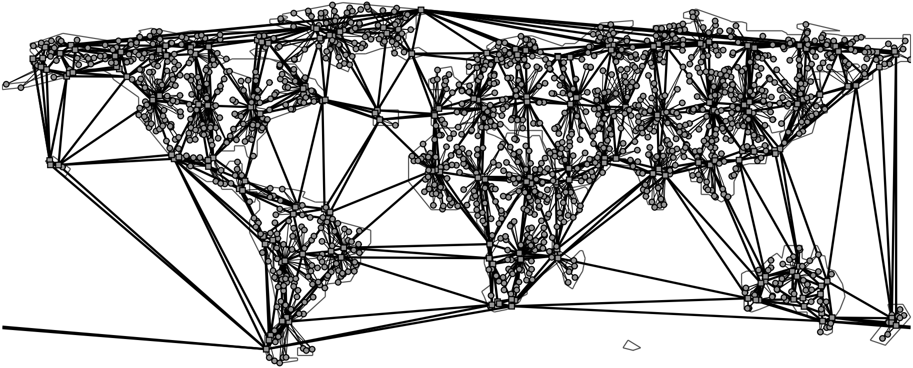


Figure: IGen topology with 1600 nodes

Test Set: Synthetic ISP Topologies [9]

Size

Name	nodes	edges	Steiner sites	terminals
IGen.1600	1600	6816	200	300
IGen.3200	3200	19410	400	600

Setup of Computational Evaluation

- 25 instances for each graph size.
- Terminate experiments after 2 hours of runtime.

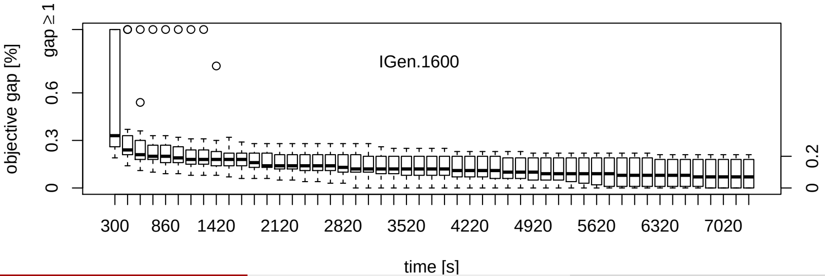
VirtuCast: Objective Gap

IGen.1600

- After 30 minutes: gap below 0.3 %
- After 120 minutes: median gap below 0.1 %

IGen.3200

- After 30 minutes: median gap around 4 %
- After 120 minutes: median gap around 3 %



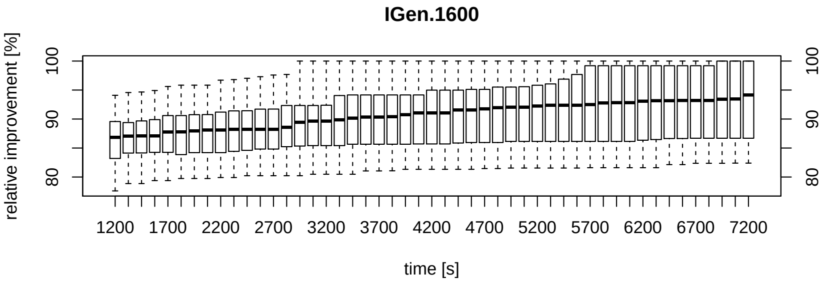
Computational Results of MCF

IGen.3200

Cannot be solved (efficiently) using MCF formulation: more than 6,000,000 variables

IGen.1600: Strength of MCF formulation

VirtuCast's lower bound improves upon MCF's lower bound by around 90% w.r.t to the best known solution.



Related Work

Molnar: Constrained Spanning Tree Problems [6]

- Shows that optimal solution is a ‘spanning hierarchy’ and not a DAG.

Oliveira et. al: Flow Streaming Cache Placement Problem [8]

- Consider a weaker variant of multicasting CVSAP without bandwidth
- Give weak approximation algorithm

Shi: Scalability in Overlay Multicasting [10]

- Provided heuristic and showed improvement in scalability.

Future Work

Model Extensions

- Generalize CVSAP for multiple concurrent multicast / aggregation sessions.
- Consider prize-collecting variants.
- Consider budgeted variants.
- Investigate usage of undirected CVSAP.

Heuristics for CVSAP

- Algorithmically challenging problem due to capacities.

Conclusion

Motivation

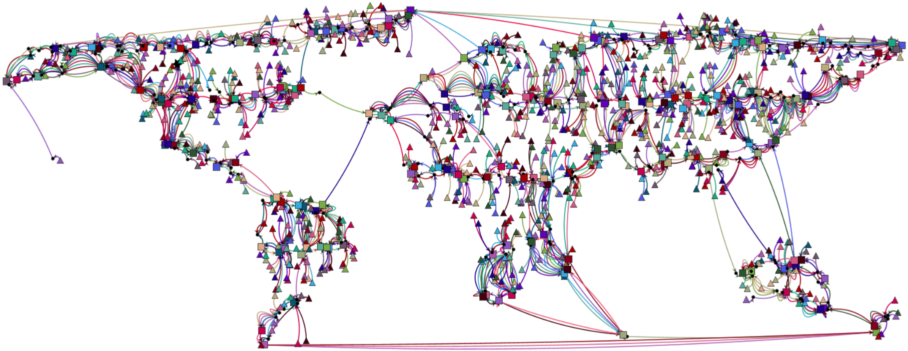
- Network virtualization enables virtual multicasting / aggregation trees.
- NFV enables placement of processing functionality.
- Goals: Improve scalability or reduce costs.

Contribution

- Concise graph theoretic definition of CVSAP.
- Algorithm to solve CVSAP: VirtuCast.
- Computational Evaluation:
  - Feasible to solve realistically sized instances using VirtuCast.
  - Significant Improvement over naive multi-commodity flow IP.

Thanks for your attention.

http://www.net.t-labs.tu-berlin.de/~stefan/cvsap.html



References I

[1] P. Costa, A. Donnelly, A. Rowstron, and G. O. Shea.
Camdoop: Exploiting In-network Aggregation for Big Data Applications.
In *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2012.

[2] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk.
Gigascope: A Stream Database for Network Applications.
In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 647–651, 2003.

[3] M. Ding, X. Cheng, and G. Xue.
Aggregation tree construction in sensor networks.
In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 4, pages 2168–2172. IEEE, 2003.
01285913.pdf.

[4] C. Hermsmeyer, E. Hernandez-Valencia, D. Stoll, and O. Tamm.
Ethernet aggregation and core network models for effcient and reliable IPTV services.
Bell Labs Technical Journal, 12(1):57–76, 2007.

[5] B. Krishnamachari, D. Estrin, and S. Wicker.
Modelling data-centric routing in wireless sensor networks.
In *IEEE infocom*, volume 2, pages 39–44, 2002.

References II

[6] M. Molnár.
Hierarchies to Solve Constrained Connected Spanning Problems.
Technical Report lrimm-00619806, University Montpellier 2, LIRMM, 2011.

[7] S. Narayana, W. Jiang, J. Rexford, and M. Chiang.
Joint Server Selection and Routing for Geo-Replicated Services.
In *Proc. Workshop on Distributed Cloud Computing (DCC)*, 2013.

[8] C. Oliveira and P. Pardalos.
Streaming Cache Placement.
In *Mathematical Aspects of Network Routing Optimization*, Springer Optimization and Its Applications, pages 117–133. Springer New York, 2011.

[9] B. Quoitin, V. Van den Schrieck, P. FranÃ§ois, and O. Bonaventure.
IGen: Generation of router-level Internet topologies through network design heuristics.
In *Proc. 21st International Teletraffic Congress (ITC)*, pages 1–8, 2009.

[10] S. Shi.
A Proposal for A Scalable Internet Multicast Architecture.
Technical Report WUCS-01-03, Washington University, 2001.