Steiner Trees Composition and Scalable Video Coding for Satelite Video Multicast

Alex Binun, Yefim Dinitz, Shlomi Dolev, Ofer Hadar, Adnan Jaber and Shevach Riabtsev

By Prof. Ofer Hadar
School of Electrical and Computer Eng.

NCA2024
Bertinoro , Italy
24-26 Oct. 2024

The research was supported by the SATELLITE project of the Israel Innovation Authority and the Rita Altura Trust chair in computer science.

Gilat

Ben-Gurion University of the Negev
Inspiration meets Excellence

Ben-Gurion University of the Negev

Research Group

Network Multimedia System (NMS) Lab

Prof. Ofer Hadar

Divya Mishra

Michael Sidorov

Shubham Agarwal

Dr. Yoram Segal

Roie Kazoom

Faina Khoroshevsky

Shachar Shmueli

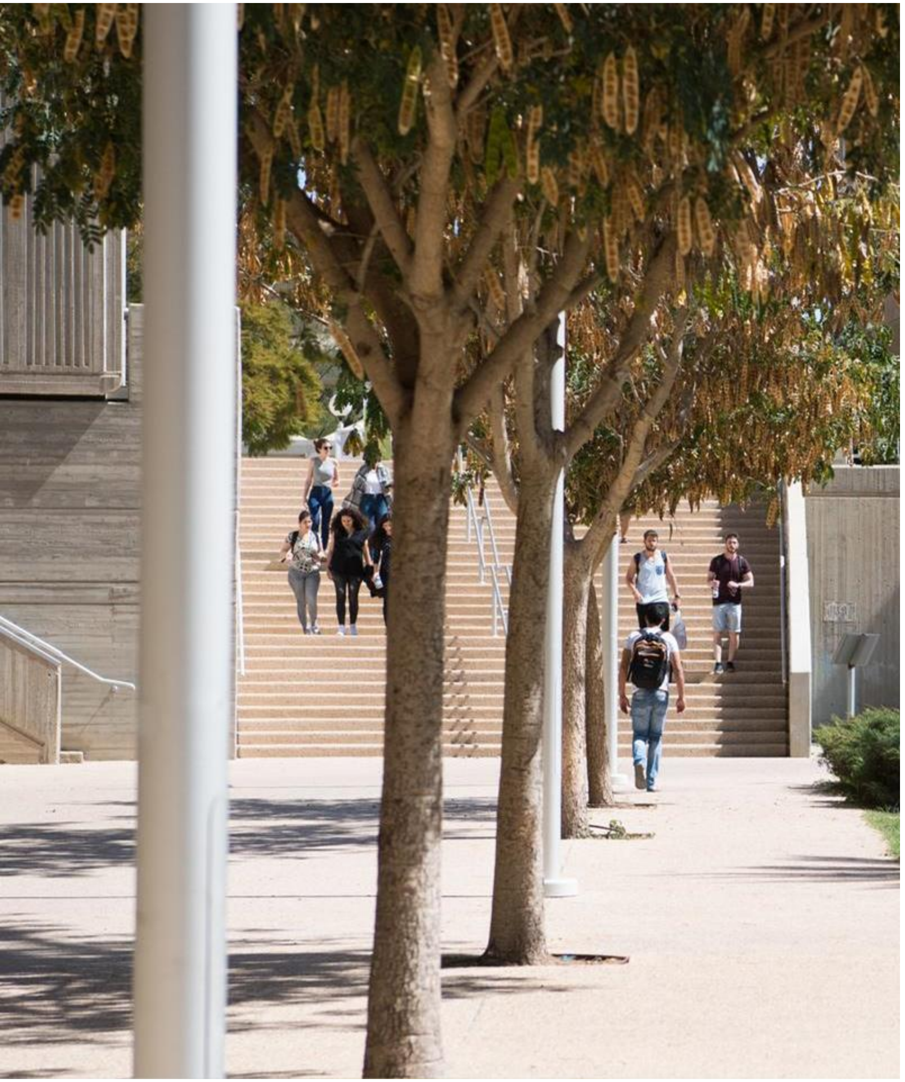
Dr. Raz Birman

Itai Dror

Ari Granevich

Ben-Gurion University of the Negev

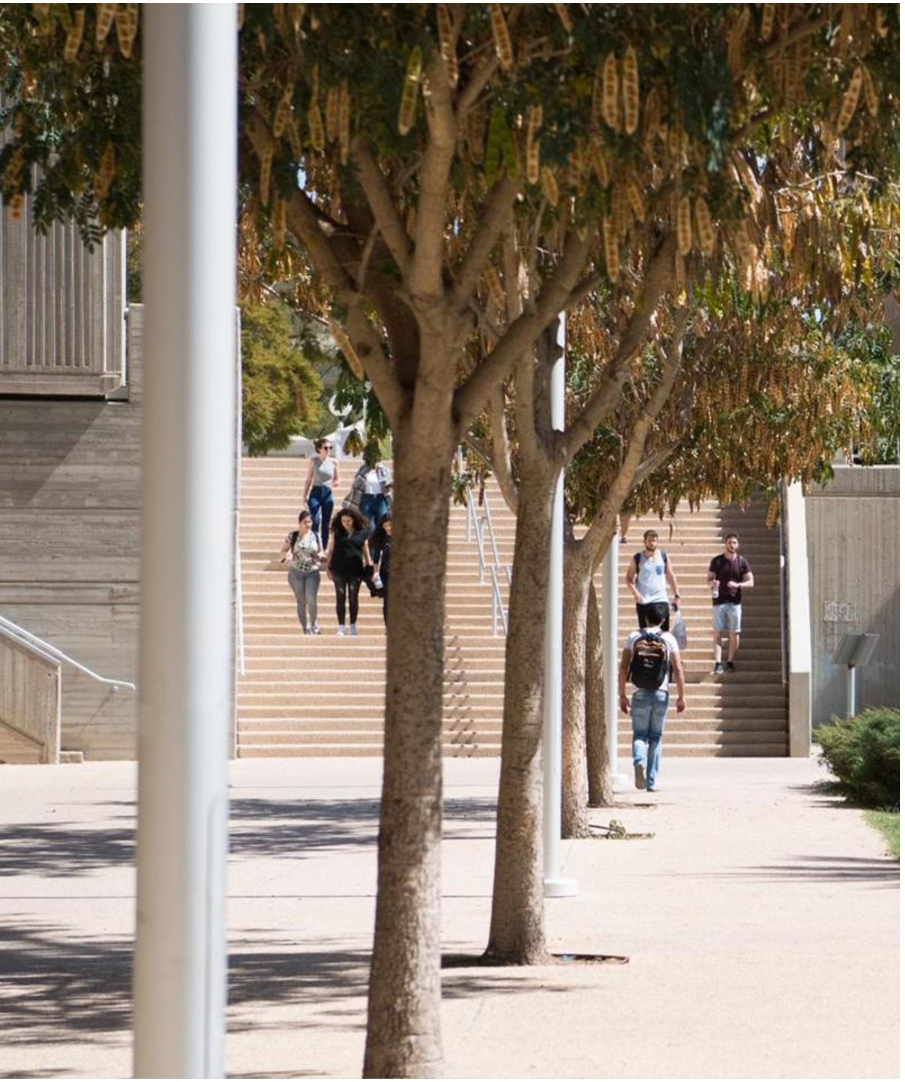
Outline



- **Recent research activity at the Lab of Network Multimedia System (NMS)**
- **Background and Motivation:** SVC coding and Multicast streaming.
- **Steiner Trees.**
- **Our algorithm :** Hierarchical Trees.
- **Simulations and Results**.
- **Conclusions**

Ben-Gurion University of the Negev

Recent research projects



- **Teleoperation of Autonomous Vehicles** (founded by the Israel Innovation Authority).
- **Adversarial AI in Vison – Trust.ai** (founded by the Israel Innovation Authority).
- **Satellites Super resolution** (founded by Ministry of Science & Technology).
- **Precision Agriculture** (founded by the Ministry of Agriculture)
- **Human body physiotherapy** (founded by Ministry of Science & Technology).
- **Video QoE from Encrypted traffic - ENTM** (founded by the Israel Innovation Authority).
- **Video compression with DNN** (founded partially by Google)
- **Deep Fake detection** (founded by Terra Incognita program).

Ben-Gurion University of the Negev

Overview

proposed scheme based on Scalable Video Coding and Optimal Steiner trees to reduce the overall traffic in the network

as the use of low Earth orbit satellites (LEO) for communication has become a reality (e.g., SpaceX). The usage of Internet communication communicating videos become very significant piece of the overall traffic

SVC Encoder

encodes a video at multiple quality layers and sends data layer by layer, so that each newly received packet will incrementally improve

Steiner trees

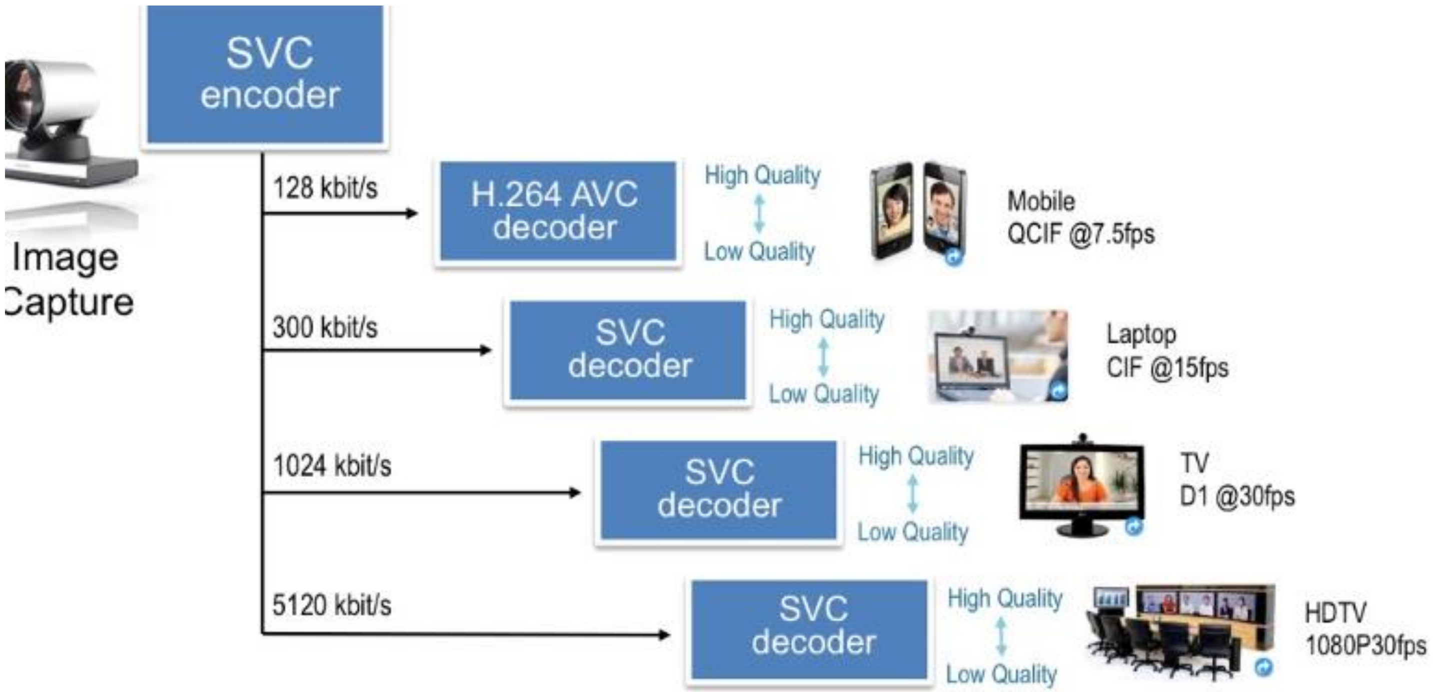
the minimum-cost tree that spans a set of given vertices in an undirected, edge-weighted graph.

Hierarchical trees

new algorithm finding Hierarchical Steiner trees , such that they are all optimal and prefer edges not used by the previous layers trees.

SVC Decoder

The decoder reconstructs the video frames from the decoded layers. It applies inverse operations to reconstruct the spatial, temporal, and quality information of the video.



Our Scheme

We suggest a new algorithm for finding the Steiner trees in the hierarchy, such that they are all optimal and prefer edges not used by the Steiner trees that are used for previous layers. Thus, distributing the communication without sacrificing optimality.

Overview

The first Steiner tree spans all the terminals and is used to convey the first layer of the SVC, and the second spans the terminals that require more resolution than the basic resolution. The third Steiner tree spans the terminals that require even more resolution, and so forth for the following Steiner trees and SVC layers.

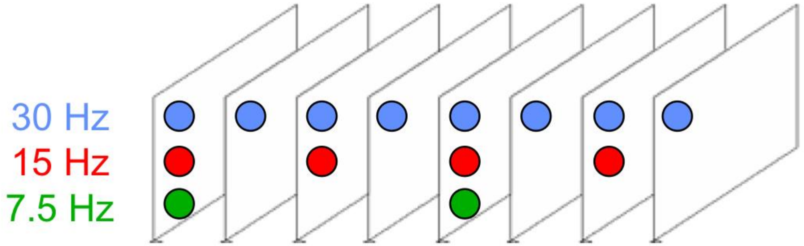
01 first optimal stiener tree that spans all terminals T1.

02 second optimal stiener tree that spans a subset of T1 called T2,

03 third optimal stiener tree that spans a subset of T2 called T3 etc..

Scalability of Video - Modalities

- Temporal: change of frame rate

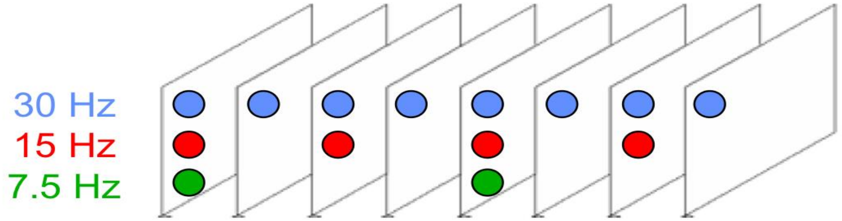


- Spatial: change of frame size

- Fidelity: change of quality (e.g. SNR)

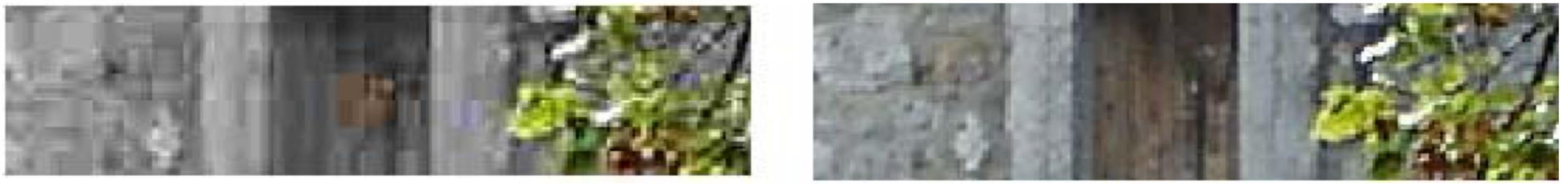
Scalability of Video - Modalities

- Temporal: change of frame rate



- Spatial: change of frame size

- Fidelity: change of quality (e.g. SNR)



SVC vs. Non-SVC

Quality	Video Resolution	SVC (Kbps) Layer only	SVC (Kbps) Total per layers	Non-SVC (Kbps) H.265	Comment
Low	480p	6,200*	6,200	5,000	Constant QP=30 was set for all streams
Medium	720p	15,000*	21,200	18,000	
High (HD)	1080p	30,000*	51,200	44,000	
4K	2160p	65,000*	116,200	99,000	

Crowd-run video source was taken from https://media.xiph.org/video/derf/
Constant QP=30 was set for all streams (good or moderate quality)

Ben-Gurion University of the Negev

SVC vs. Non-SVC contribution

- Bitrate saming can potenitialy range from **around 10% to 40%** .
- It depends of the distribution of users with different requrements.
- For many distributions, using SVC pays off :
  - In our case: (25%, 25%, 25%, 25%) and (0%, 40%, 30%, 20%).
  - The saving of SVC with compared to Non-SVC was between 15%-25% of the traffic.

Ben-Gurion University of the Negev

Example of SVC with Spatial video Scalability

Only one layer : Base layer video

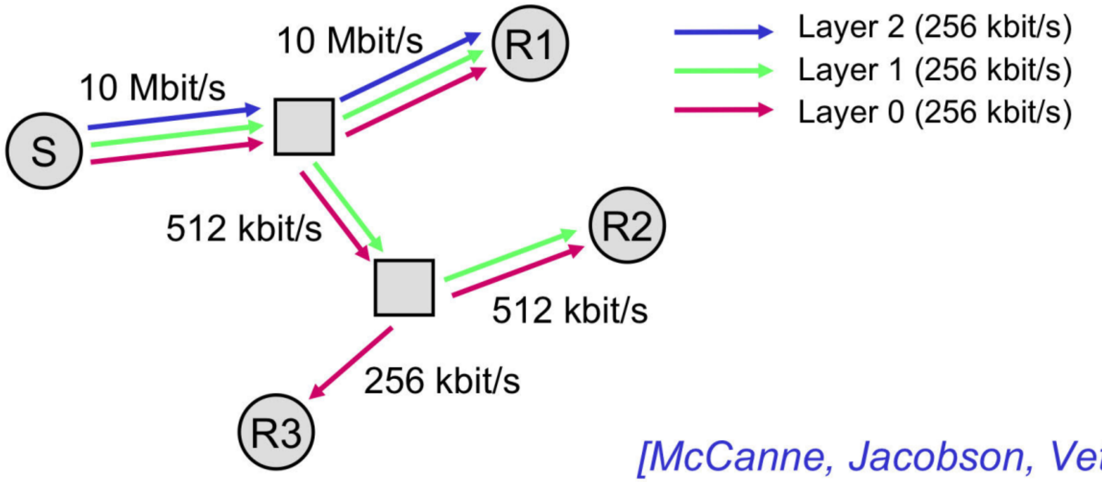
Two layers video : Base layer + Enhacement layer 1

Example of SVC with spatial video scalability (cont.)

Three layers video : Base layer + Enhacement layer 1 + Enhacement layer 2

Receiver-Driven Layered Multicast

- Video and audio are encoded using layered, scalable scheme
- Different layers are transmitted on different multicast groups
- Each receiver subscribes to the base layer and depending on the available data rate to one or more enhancement layers
- Adaptation is carried out by joining or leaving groups



[McCanne, Jacobson, Vetterli, 96]

Streaming multimedia: DASH

- *DASH: D*ynamic, *A*daptive *S*treaming over *H*TTP
- *server:*
  - divides video file into multiple chunks
  - each chunk stored, encoded at different rates
  - *manifest file:* provides URLs for different chunks
- *client:*
  - periodically measures server-to-client bandwidth
  - consulting manifest, requests one chunk at a time
    - chooses maximum coding rate sustainable given current bandwidth
    - can choose different coding rates at different points in time (depending on available bandwidth at time)

Ben-Gurion University of the Negev

Visualization of Hierarchical trees using HST vs. No using of HST

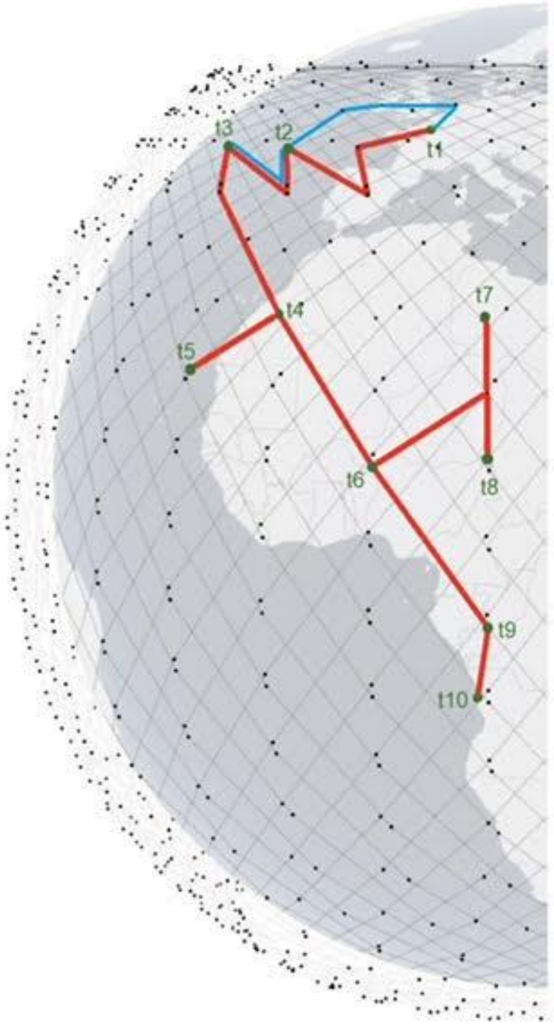


Figure 8 Visualization of hierarchical trees using HST

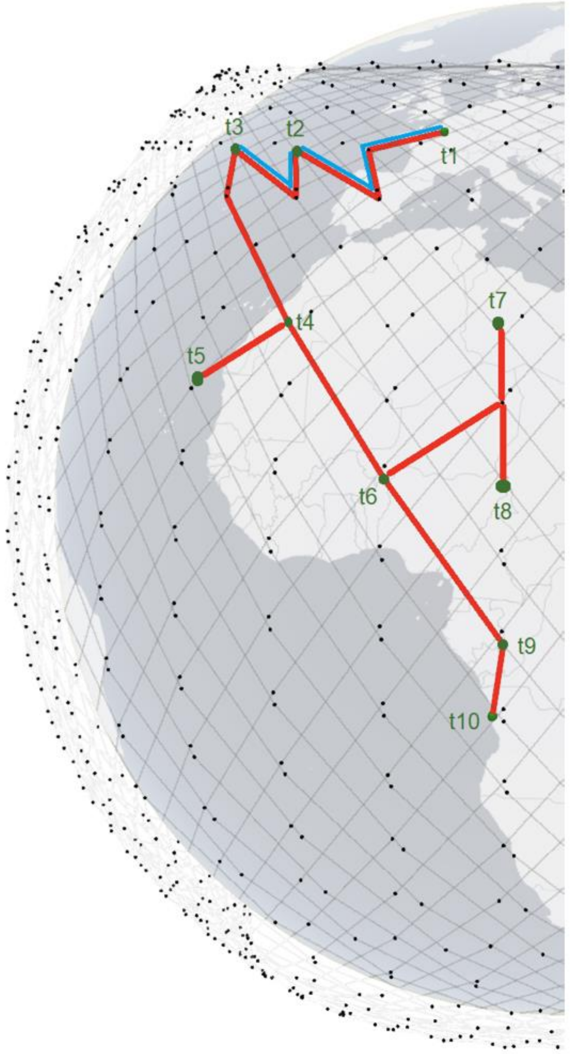


Figure 9 Visualization of hierarchical trees without using HST

Ben-Gurion University of the Negev

Steiner tree & Heuristic Algorithms

The Steiner tree problem is one of the best-studied problems in Computer Science. Given a connected graph G = (V,E) on n = |V | nodes, edge weights w: E– –→ R+, in fact, we restrict the weights here to be positive integers and a set S ⊆ V of k terminals, the objective is finding a subtree ST of G spanning S such that the weight w(ST) of ST is minimized.

Heuristic algorithms are commonly used to find near-optimal solutions within a reasonable amount of time. The SPG has seen numerous theoretical advances in the last 10 years, bringing forth significant improvements in complexity and approximability.

The Prim-Dual algorithm for the Steiner Tree Problem is polynomial in terms of time and computation complexity and it provides a guaranteed approximation ratio. It may not always produce the exact optimal solution, but it often provides reasonably good solutions efficiently. Approximation algorithms are used at the initial stage of the solver to decrease the problem size.

Ben-Gurion University of the Negev

overview

The Steiner tree problem is one of the best-studied problems in Computer Science. Given a connected graph $G = (V,E)$ on $n = |V|$ nodes, edge weights $w: E \dashrightarrow R+$, in fact, we restrict the weights here to be positive integers and a set $S \subseteq V$ of k terminals,
the objective is finding a subtree ST of G spanning S such that the weight w(ST) of ST is minimized.

Based on that we suggest a new algorithm for finding the Steiner trees in the hierarchy HST.

HST Algorithm

Require: Graph $G = (V,E,w)$ and sets of terminals $S1, S2, \ldots Sn$
Ensure: Sequence of n optimal Steiner trees STi with terminals sets Si, $1 \leq i \leq n$, with the hierarchically minimal overlapping on edges
for each $e \in E$ do
$w'(e) \leftarrow w(e) \cdot |V|$
end for
for each i from 1 to n do
Find an optimal Steiner tree STi in graph $G = (V,E,w')$ with the terminals set Si
for each $e \in STi$ do
if $e \in STi$ AND $e \notin STj$, $j \leq i - 1$ then
$w'(e) \leftarrow w(e) + 1$
end if
end for
end for
return (STi), $1 \leq i \leq n$

Ben-Gurion University of the Negev

PROOFS

Lemma 3.1 Given a weighted graph G = (V,E,w) with a set of terminals S, the following, relatively slight change of edge weights does not affect Steiner tree optimality, in the following sense. If we change all the edge weights by multiplying them by |V | and, after that, adding 1 for some of the edges, then any Steiner tree for G and S optimal with respect to the new weights is optimal also with respect to the original weights.

Proof. Consider the new weighted graph G′ = (V,E,w′), where w′(e) is either w(e) · |V | or w(e) · |V |+1, for each edge e ∈ E. Let ST be an optimal Steiner tree for G and S. Denote the weight of ST in G by X. Then, the weight of ST in G′ is at most X · |V | + |V | − 1.

Assume, by contradiction, that an optimal Steiner tree ST′ for G′ and S is not optimal in G. This implies that the weight of ST′ in G is at least X+1, so that the weight of ST′ in G′ is at least (X+1)· |V |. This contradicts the existence of Steiner tree ST for S whose weight in G′ is at most X · |V | + |V | − 1 < (X + 1) · |V |.

Theorem 3.2 Algorithm 1 finds a sequence of hierarchically optimal Steiner trees.

Proof. Each Steiner tree STi found by Algorithm 1 is optimal for G and Si by Lemma 3.1.

Note that Algorithm 1 permanently maintains that for every edge e, its weight w′(e) is either w(e) · |V | + 1, if e is an edge of at least one previously constructed Steiner tree, or w(e) · |V |, otherwise. Consider any iteration i; denote w(STi) by Xi. The weight w′(ST) of any Steiner tree optimal for G and Si is Xi · |V | + Y (ST), where Y (ST) is the number of edges in ST overlapping with the previously found trees STj , j ≤ i − 1. Since the ith iteration of Algorithm 1 finds Steiner tree STi minimal with respect to weight w′ and since Xi · |V | is a constant, STi minimizes Y (ST), as required.

PROOFS

overview

The load balancing generalization of the previous problem setting. As previously, we require that each Steiner tree STi be optimal. Besides, we will aim to equalize the edge loads in the following sense. Let us say that the load on edge e is k if e is used by exactly k Steiner trees. The goal is:

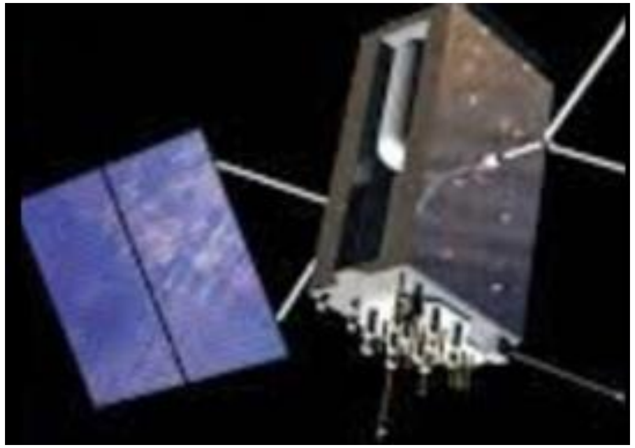
- to minimize the maximal load,
- while fixing it, to minimize the number of edges with that load,
- while fixing the above, to minimize the second maximal load,
- while fixing the above, to minimize the number of edges with that load,
- and so on.

LBHST Algorithm

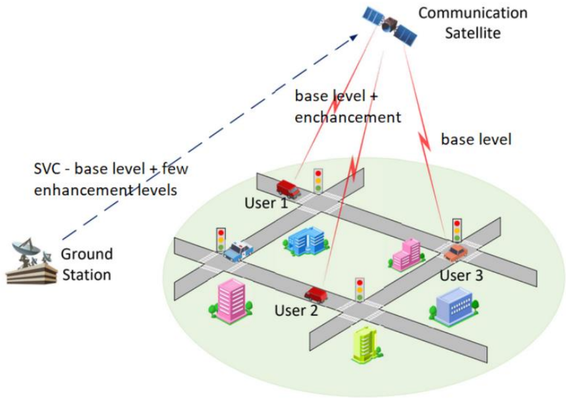
Require: Graph G = (V,E,w) and sets of terminals S1, S2, . . . Sn
find optimal Steiner tree ST1 for G = (V,E,w) and S1
for (i = 2 to n) do
for each ($e \in E$) do
$w'(e) \leftarrow w(e) \cdot |V|^{i-1}$
compute #(e): the number of times that e appears in trees Sj , j < i
end for
for each ($e \in E$) do
if #(e) > 0 then
$w'(e) \leftarrow w'(e) + |V|^{\#(e)-1}$
end if
end for
find optimal Steiner tree STi for G' = (V,E,w') and Si
end for
return (ST1, ST2, ..., STn)

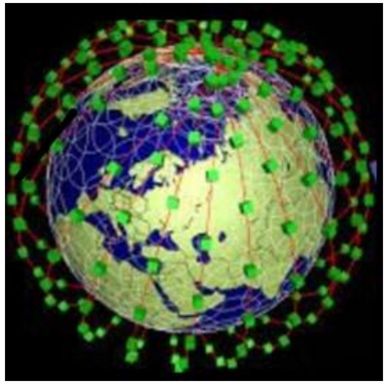
Ben-Gurion University of the Negev

Simulation and Experiments : Visualization



- Hypatia – calculates satellite orbits and renders them
- Satellite specifications (as TLEs) are loaded from the configuration file
- Satellite orbits are calculated using the library AstroPy
- Satellite trajectories are rendered :
  - In a web application: using the Cesium library
  - In the standalone NS-3 network simulator



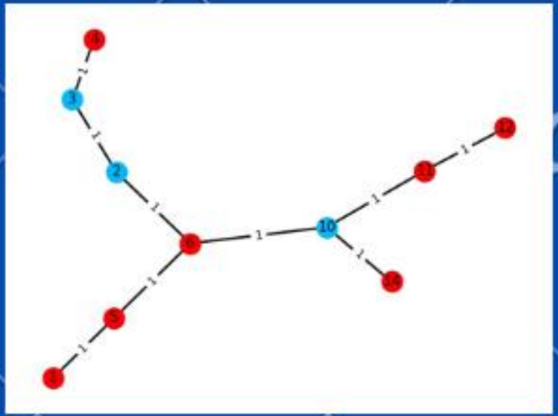


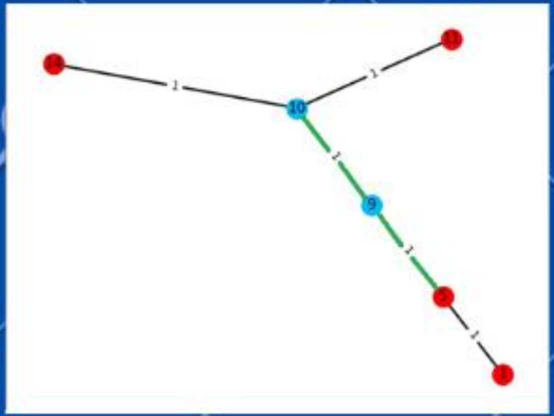
Ben-Gurion University of the Negev

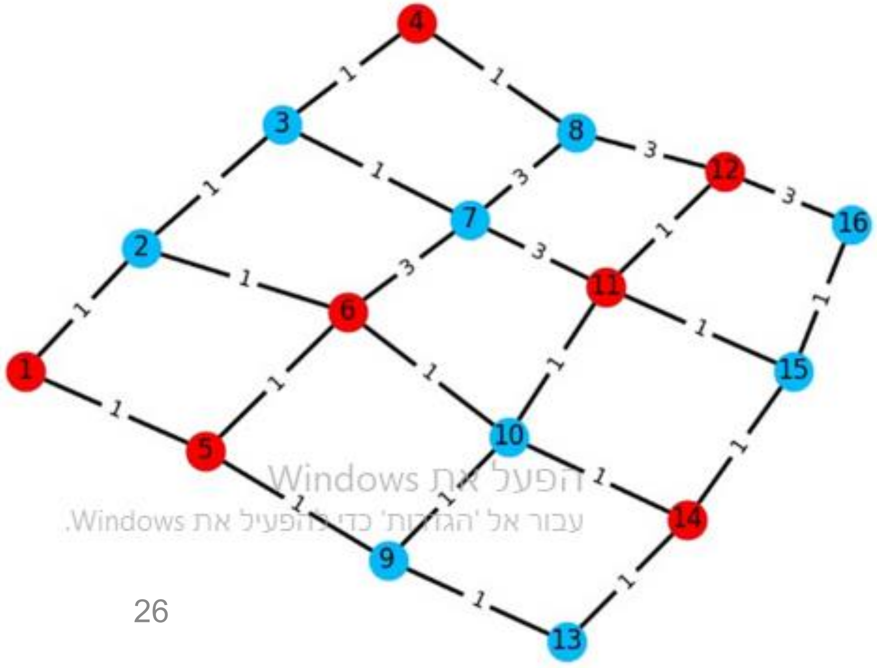
Ben-Gurion University of the Negev

EXPERIMENT

Make it easier to gather and share information







Improve Communication

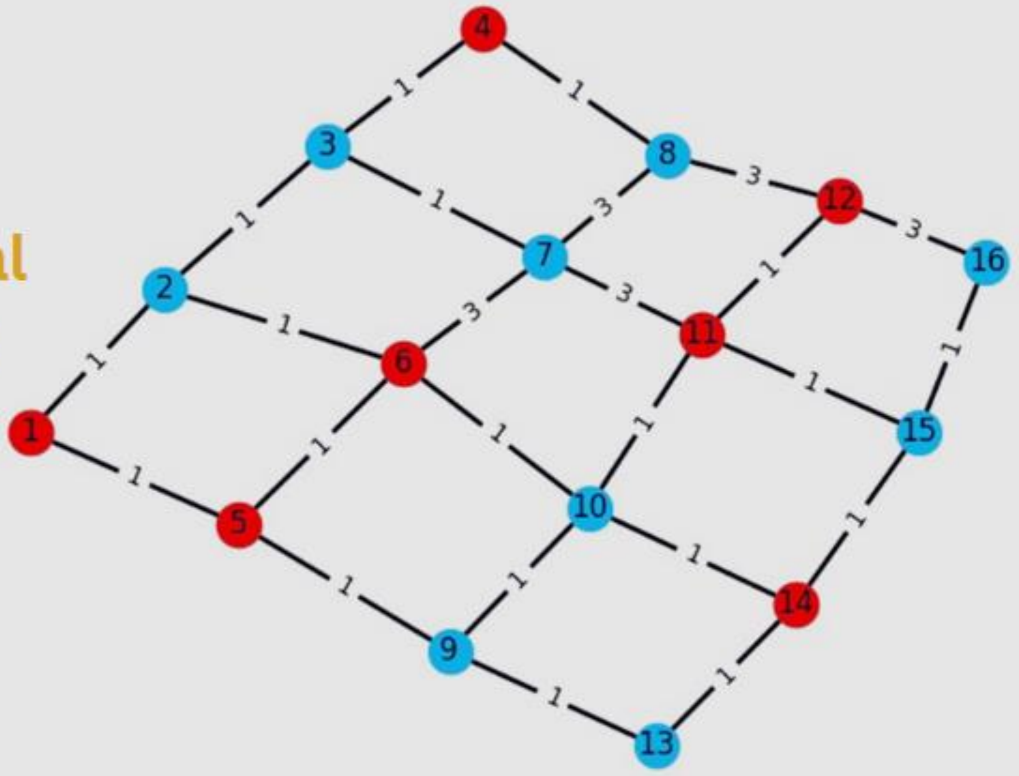
for 4x4 mesh grid shown on the left and a set of terminals T1={1,4, 5, 6, 11, 12, 14 } and T2={1, 5, 11, 14) marked in red the two optimal hierarchical Steiner trees are shown above.

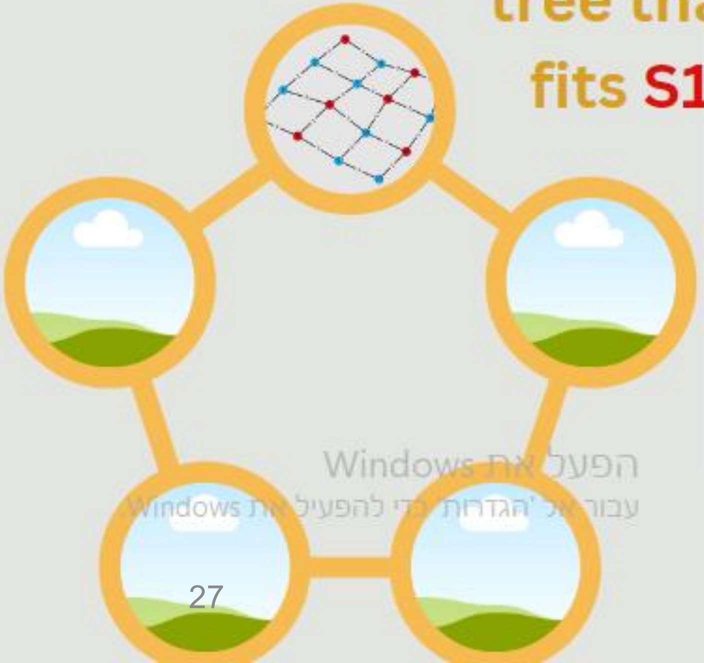
while the left tree weight is 9 and fits T1 and the right tree weight is 5 and it fits T2. Through the corresponding optimal trees, we deliver SVC layers, the base SVC layer is delivered through the first hierarchical tree, and the first svc enhancement layer is delivered through the second hierarchical tree, etc...

ALGORITHM

Step 1:

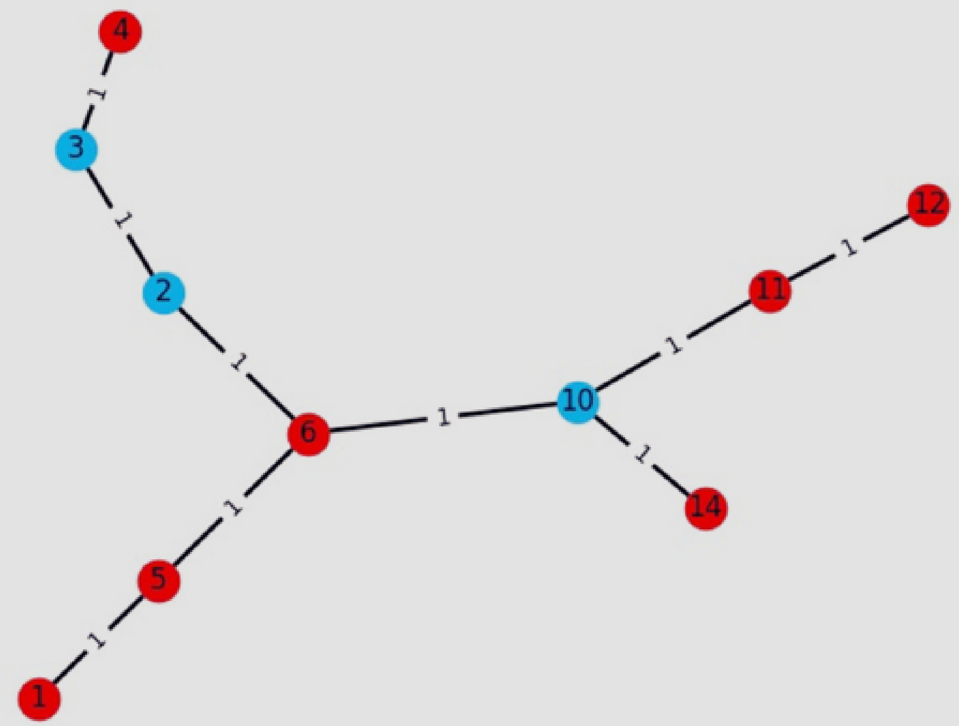
find the first hierarchical tree that fits S1



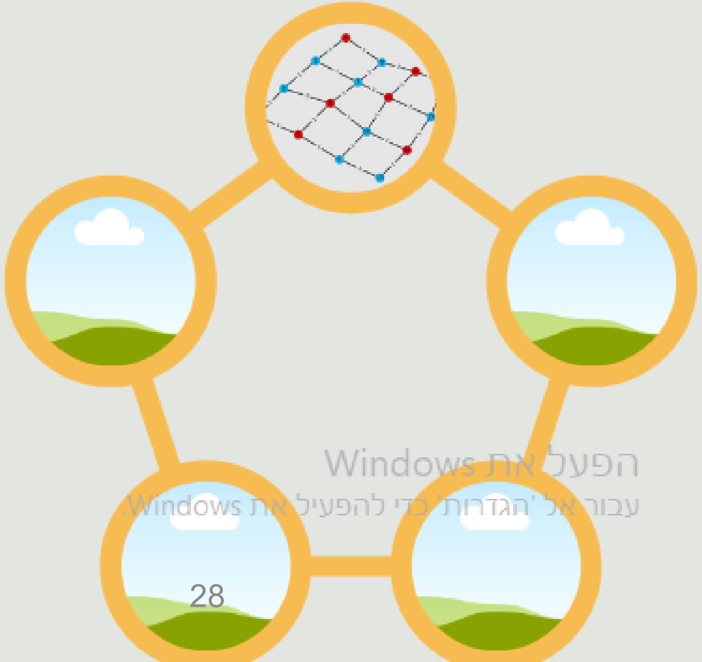


Ben-Gurion University of the Negev

ALGORITHM

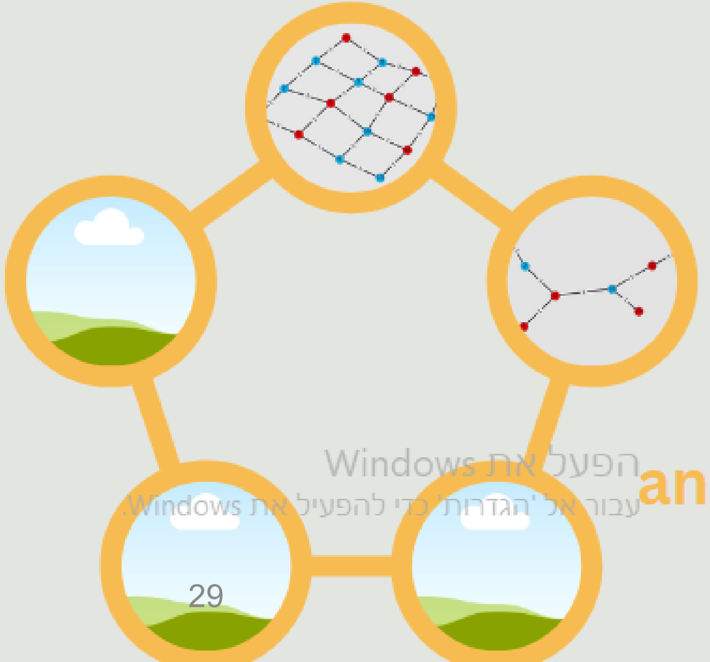


first hierarchical tree



ALGORITHM

Step 2:

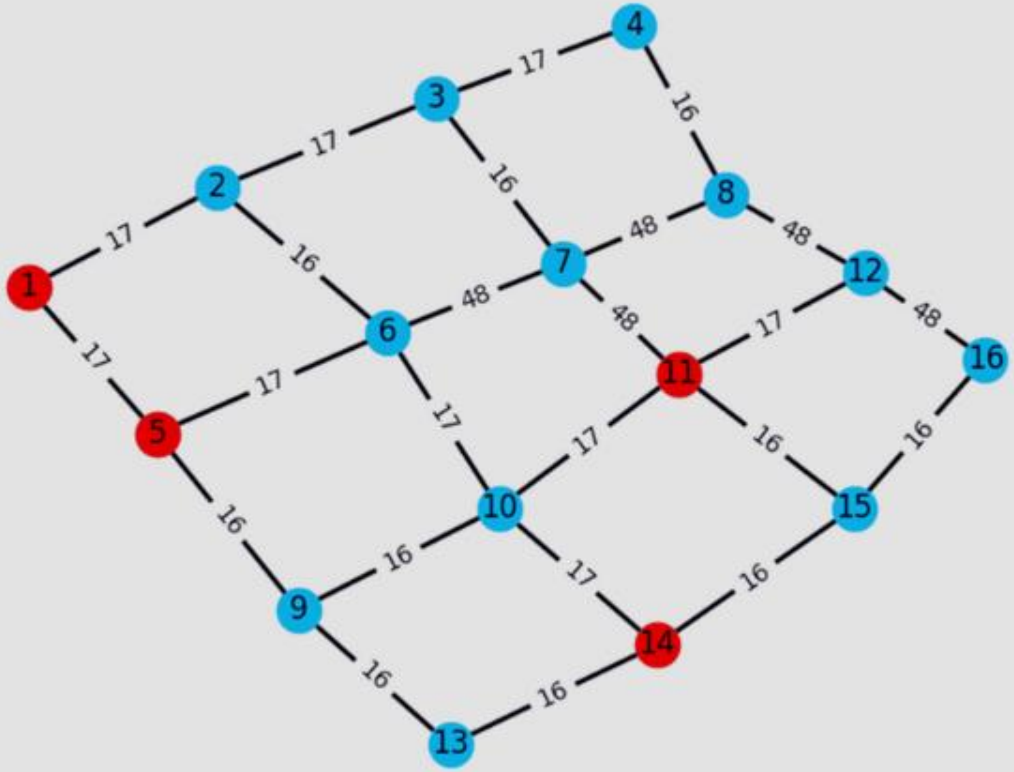


multiply each edge by |v|

and add 1 to each edge chosen in the first tree

Ben-Gurion University of the Negev

ALGORITHM

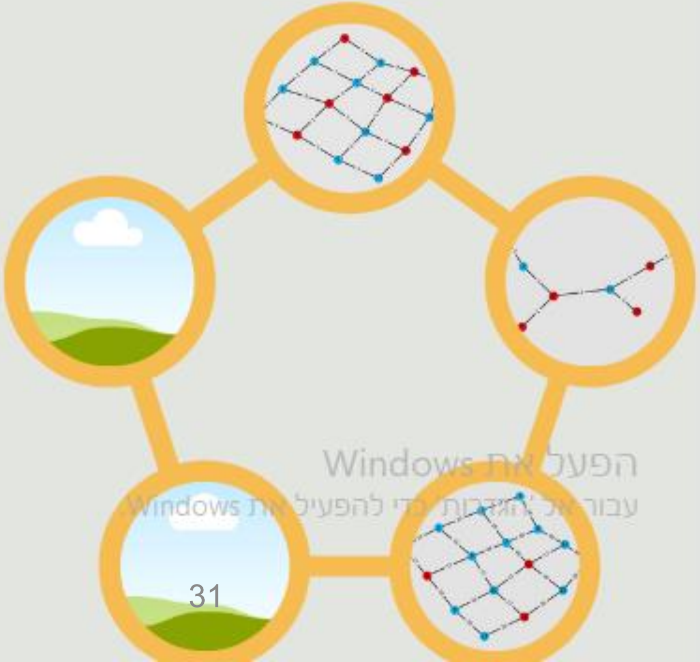


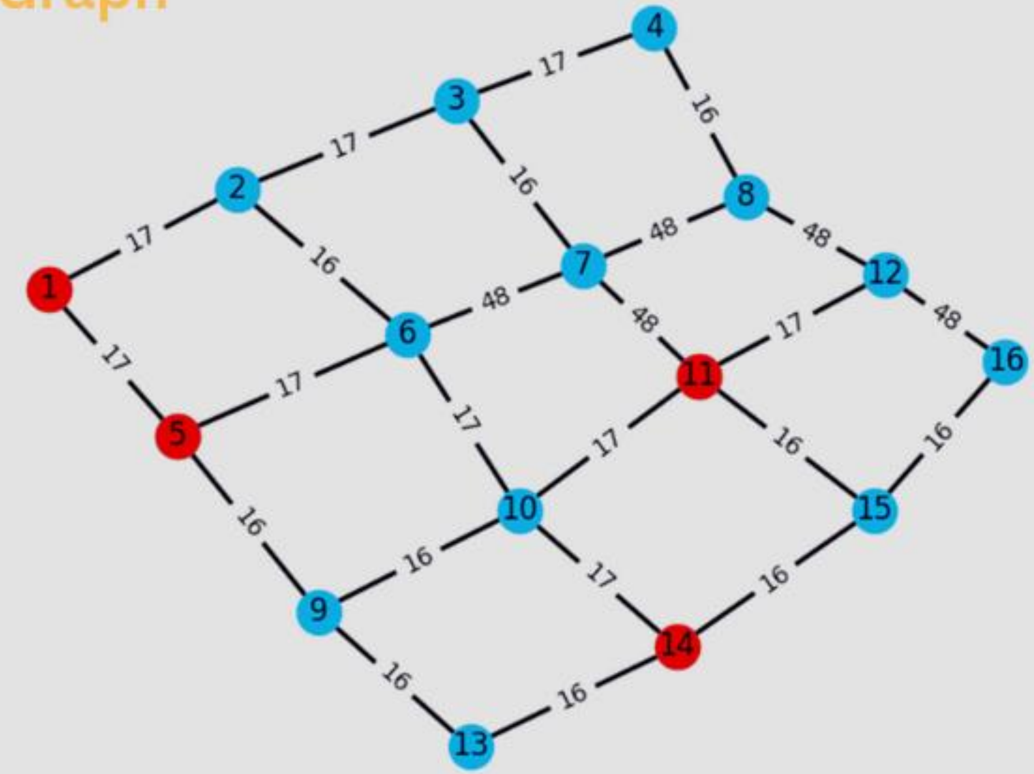
Ben-Gurion University of the Negev

ALGORITHM

Step 3:

Find the second hierarchical tree fits S2 in the new weighted Graph

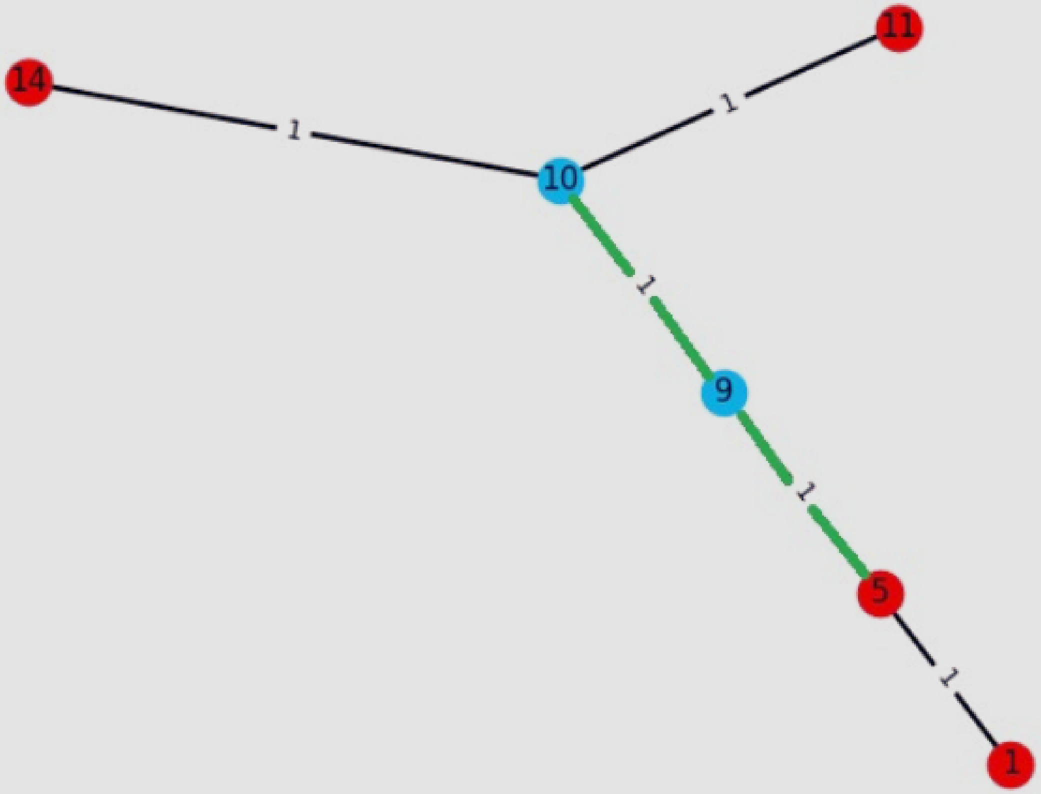


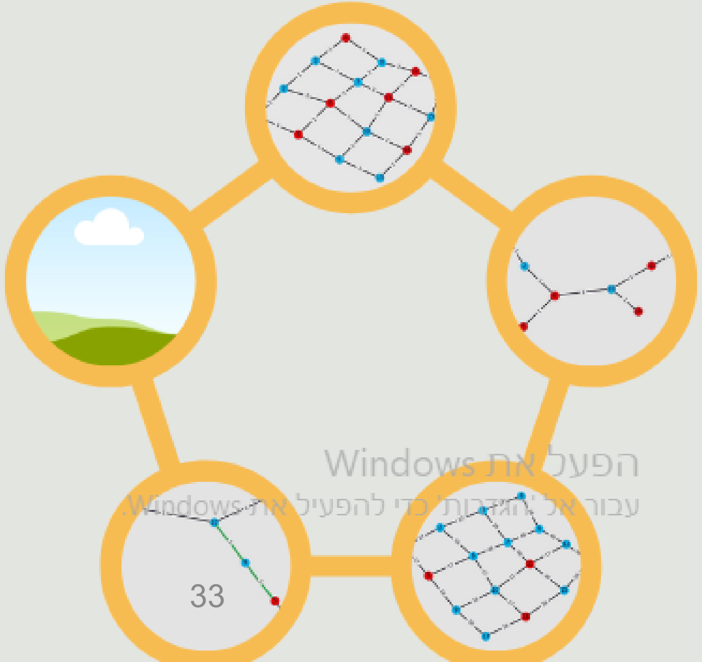


Ben-Gurion University of the Negev

RESULTS

HST produces two optimal hierarchical trees and used two distinct edges marked in green to construct the second hierarchical tree.

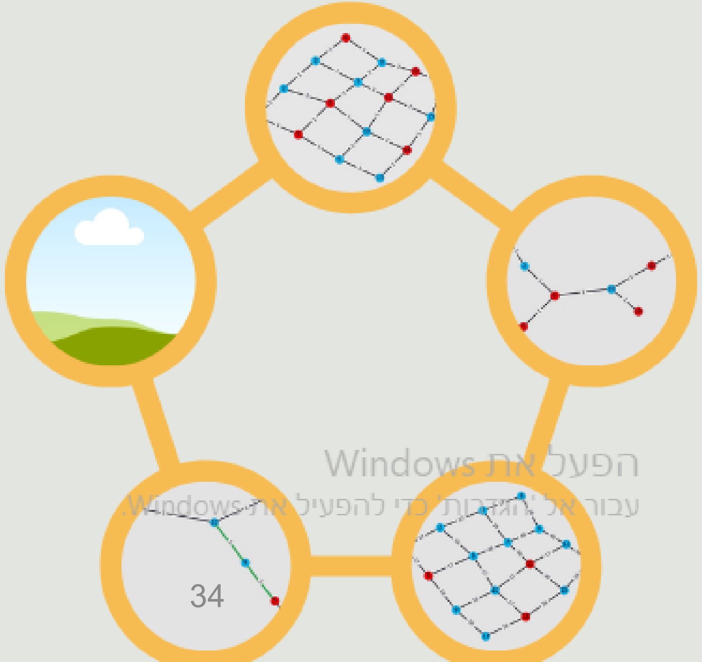


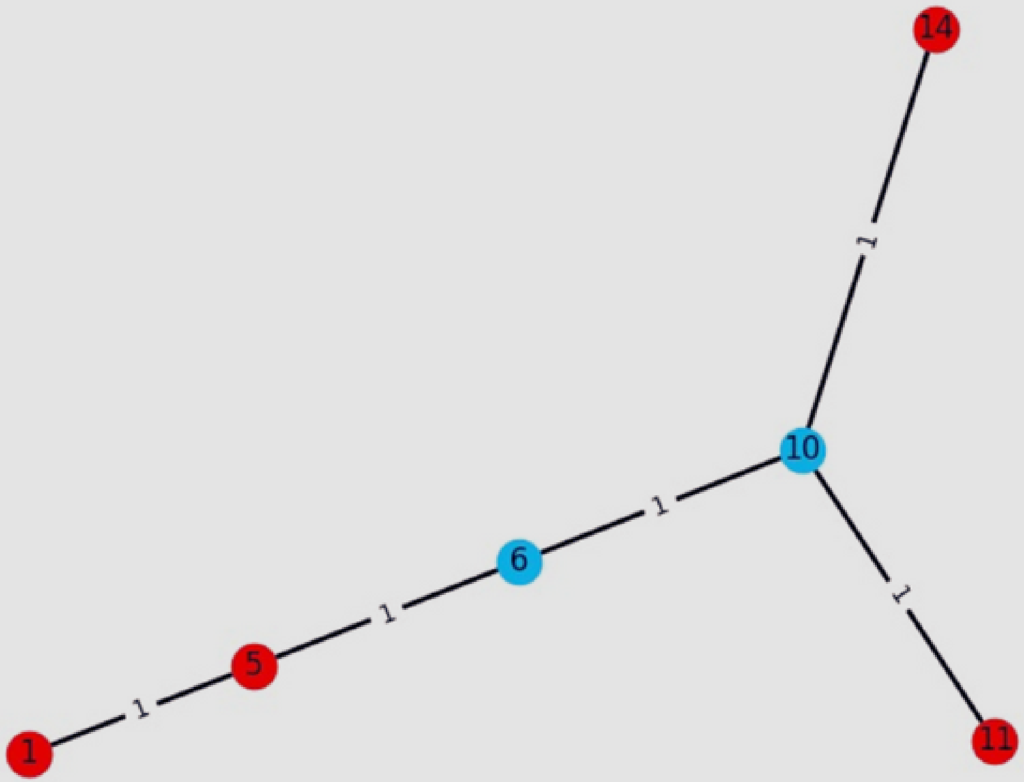


Ben-Gurion University of the Negev

RESULTS

while without applying HST on the graph all the edges used to construct the first tree used to construct the second hierarchical

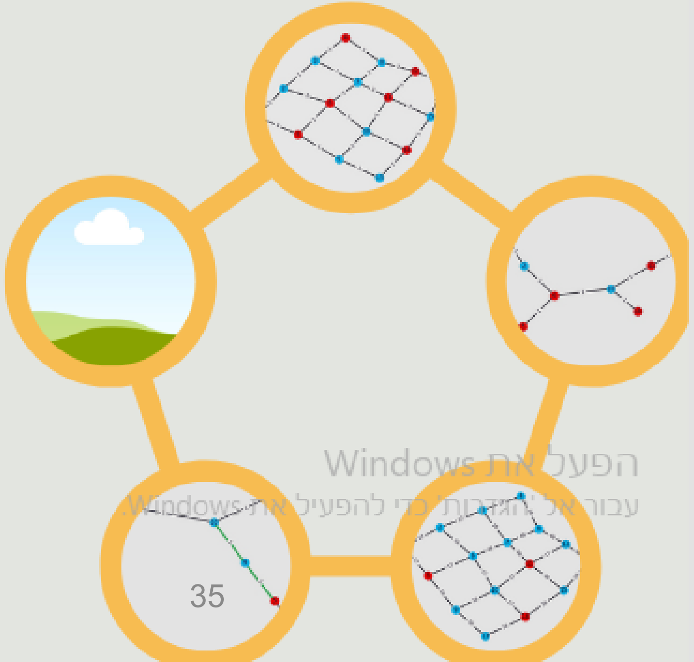


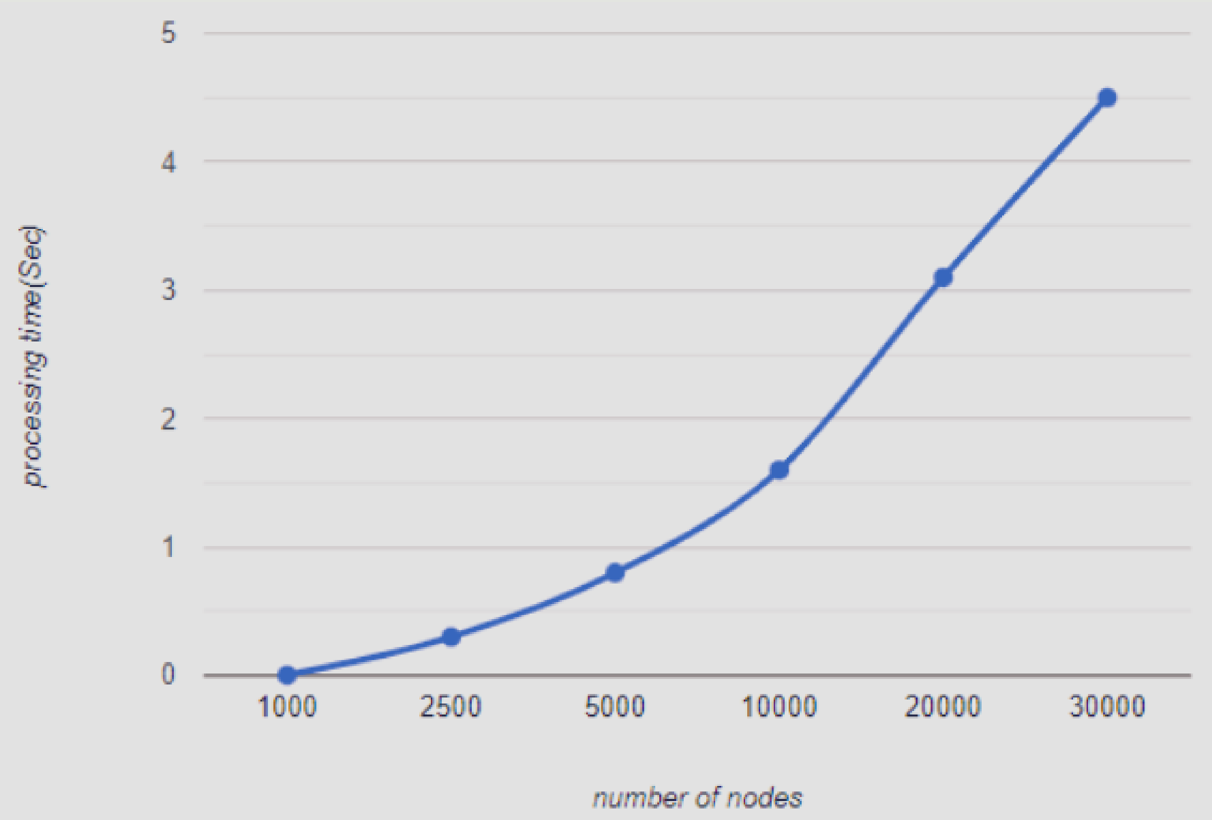


Ben-Gurion University of the Negev

PERFORMANCE

worse case processing time finding hierarchical trees for any set of terminals





Ben-Gurion University of the Negev

Conclusions

1. Optimizing Bandwidth in Video Delivery:

- Splitting the bit stream into layers and recombining them at the client end optimizes bandwidth for various network conditions.
- Experiments show that encoding strategies and video content significantly influence bit stream savings.

2. Hierarchical Steiner Tree (HST) Algorithms in Network Optimization:

- HST algorithms, implemented using the SCIP-Jack solver, efficiently discover optimal routes in complex networks.
- Their efficacy is particularly noticeable in smaller graphs, reducing edge duplication and improving resource use.

3. Fault Tolerance:

- The solution is resilient to faults, ensuring that video delivery and network configurations maintain performance under failure conditions, improving reliability across diverse scenarios.

4. Application Areas:

- Video delivery (using SVC) and network planning (using HST algorithms) offer adaptable, efficient, and fault-tolerant solutions.
- Potential applications include communication networks, transportation, and logistics.

5. Broader Impact:

- These findings contribute to multimedia technology and network optimization, laying the groundwork for future research and practical implementations across diverse industries.

Prof. Shlomi Dolev : shlomi@cs.bgu.ac.il

Prof. Ofer Hadar : hadar@bgu.ac.il

THANK YOU

Ben-Gurion University of the Negev

Ben-Gurion University of the Negev