Graph-Model-Based Dynamic Routing and Spectrum Assignment in Elastic Optical Networks[1]

Presenter: Yongcheng(jeremy) Li
PhD student, School of Electronic and Information Engineering,
Soochow University, China
Email: liyongcheng621@163.com

Group Meeting, Friday, July 7, 2016

[1] C. F. Hsu, Y. C. Chang, S. C. Sie, “Graph-Model-Based Dynamic Routing and Spectrum Assignment in Elastic Optical Networks,” Journal of Optical Communications and Networking, vol. 8, no. 7, pp. 507-520, Jun. 2016.

Outline

1. Background
2. Layered Graph Model
3. Performance evaluation
4. Conclusions

Background

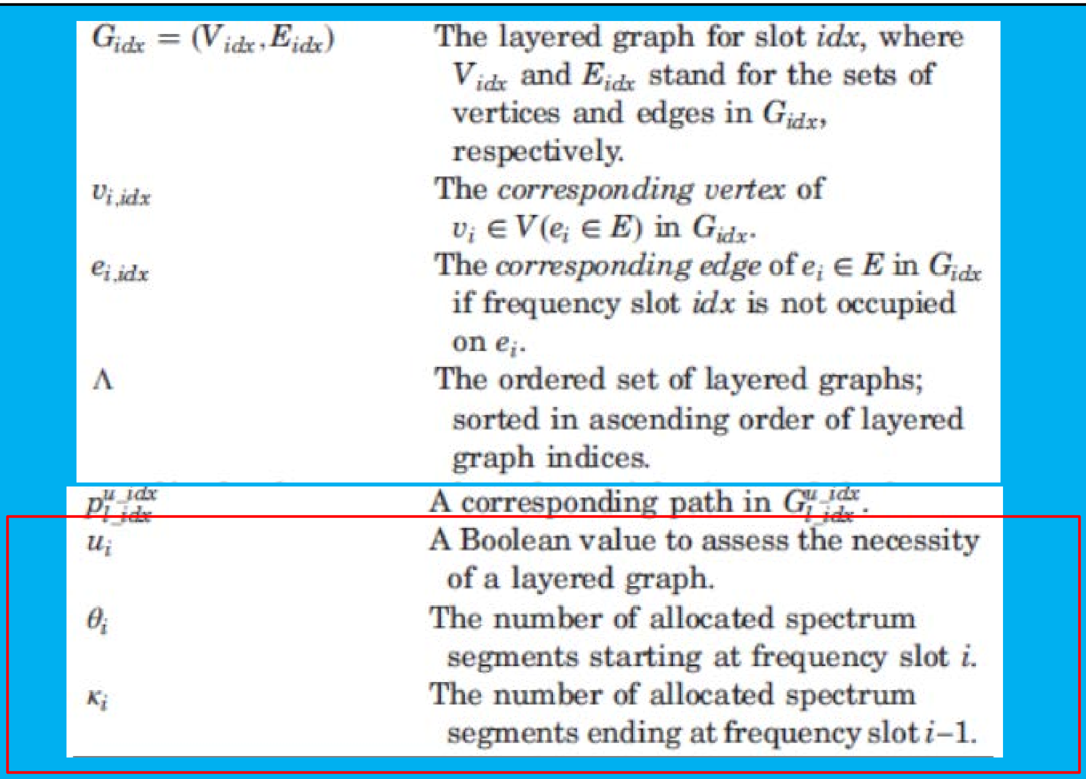
- In recent years, the emergence of coherent optical orthogonal frequency-division multiplexing (CO-OFDM) technology eventually enhanced spectrum utilization efficiency and realized a flexible resource allocation mechanism
- A novel network architecture called a spectrum-sliced elastic optical path network (SLICE) is proposed
- Recent literature of this field also refers to this framework as elastic optical networks (EONs)
- Compared with RWA problem in WDM networks, the resource allocation problem in EONs evolves into the (RSA) problem because of finer basic switching granularity
- The constraints of the RSA problem include spectrum continuity, spectrum non-overlapping, and spectrum contiguity

Layered Graph Model

- Because of *fine grid* and *unique spectrum contiguity* constraint, extending traditional layered graph model from wavelength level to subcarrier level is not an intelligent approach
- Propose an enhanced layered graph model to deal with the RSA problem while reducing both time and space complexity
- This model is suitable for both grid and gridless spectrum standards
- Modify original layered graph model to filter out those links with at least *one available segment* to accommodate the request
- Exploit the layered graph model to generate an auxiliary graph, which is called a *filter graph* in this work, and then apply some routing scheme on this graph

Layered Graph Model

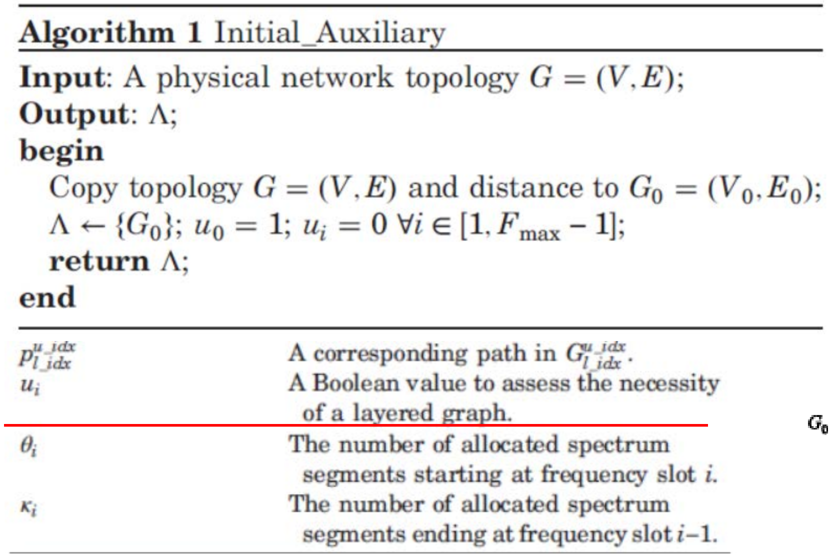
- Notations:



$G_{idx} = (V_{idx}, E_{idx})$	The layered graph for slot *idx*, where V_{idx} and E_{idx} stand for the sets of vertices and edges in G_{idx}, respectively.
$v_{i,idx}$	The *corresponding vertex* of $v_i \in V (e_i \in E)$ in G_{idx}.
$e_{i,idx}$	The *corresponding edge* of $e_i \in E$ in G_{idx} if frequency slot *idx* is not occupied on e_i.
Λ	The ordered set of layered graphs; sorted in ascending order of layered graph indices.
$p_{l\_idx}^{u\_idx}$	A corresponding path in $G_{l\_idx}^{u\_idx}$.
u_i	A Boolean value to assess the necessity of a layered graph.
θ_i	The number of allocated spectrum segments starting at frequency slot *i*.
κ_i	The number of allocated spectrum segments ending at frequency slot *i*−1.

Layered Graph Model

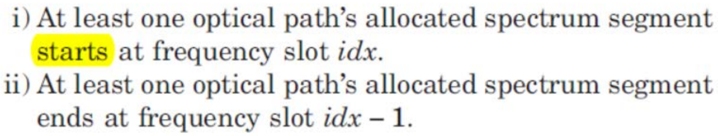
- Create layered graph



Algorithm 1 Initial_Auxiliary

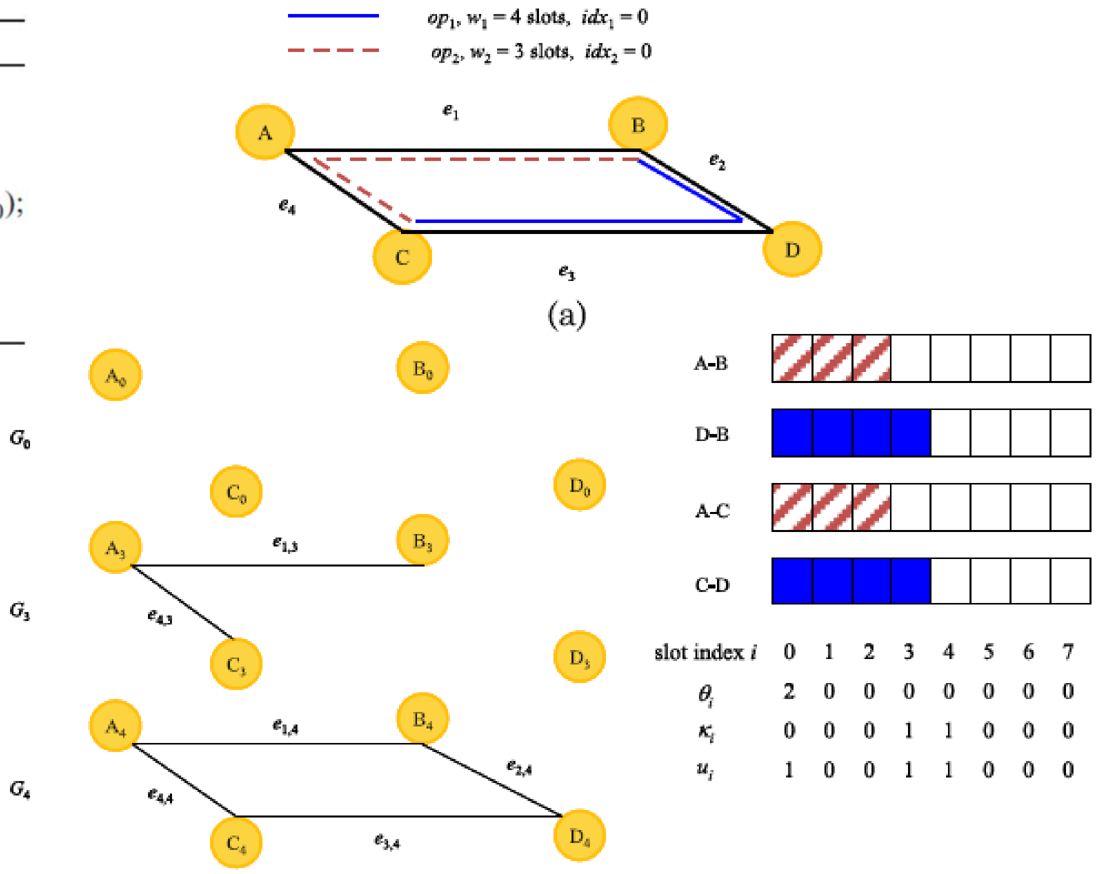
```
Input: A physical network topology $G = (V, E)$;
Output: $\Lambda$;
begin
    Copy topology $G = (V, E)$ and distance to $G_0 = (V_0, E_0)$;
    $\Lambda \leftarrow \{G_0\}$; $u_0 = 1$; $u_i = 0 \ \forall i \in [1, F_{\max} - 1]$;
    return $\Lambda$;
end
```

$p_{l\_idx}^{u\_idx}$	A corresponding path in $G_{l\_idx}^{u\_idx}$.
u_i	A Boolean value to assess the necessity of a layered graph.
θ_i	The number of allocated spectrum segments starting at frequency slot i.
κ_i	The number of allocated spectrum segments ending at frequency slot i–1.



i) At least one optical path's allocated spectrum segment starts at frequency slot *idx*.

ii) At least one optical path's allocated spectrum segment ends at frequency slot *idx* – 1.



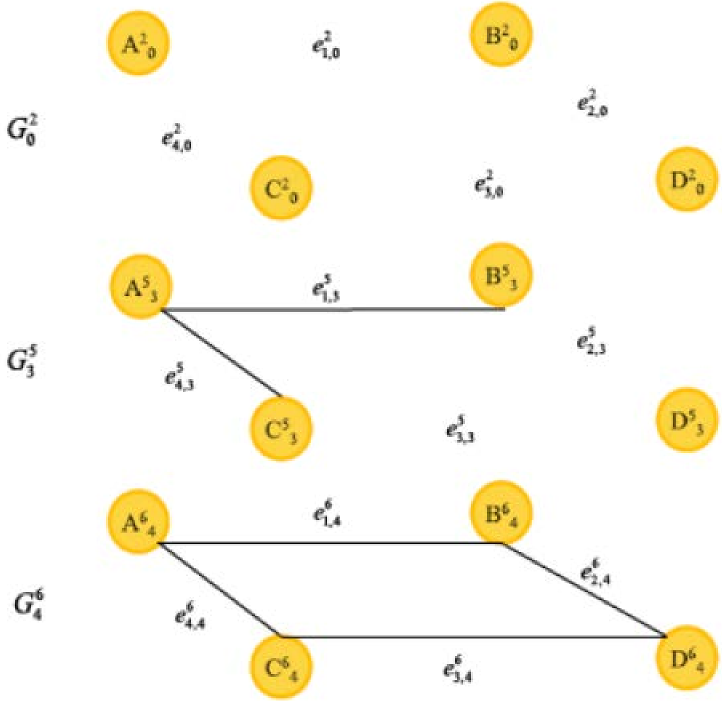
slot index i	0	1	2	3	4	5	6	7
θ_i	2	0	0	0	0	0	0	0
κ_i	0	0	0	1	1	0	0	0
u_i	1	0	0	1	1	0	0	0

Layered Graph Model

- Filter graph
  - A request with demand volume W_n is provisionable on a spectrum segment starting from slot l_idx is equivalent to finding a path on the filter graph $\boldsymbol{G_{l\_idx}^{l\_idx+W_n-1}}$

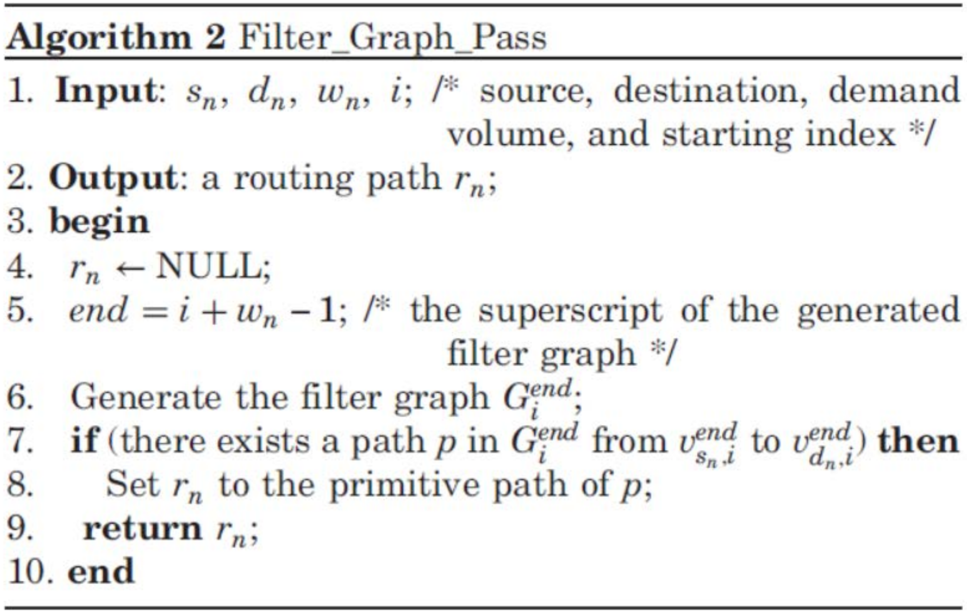
$$\boldsymbol{v_i} \quad \Longrightarrow \quad \boldsymbol{v_{i,l\_idx}^{u\_idx}}$$

$$\boldsymbol{e_i} \quad \Longrightarrow \quad \boldsymbol{e_{i,l\_idx}^{u\_idx}}$$



Layered Graph Model

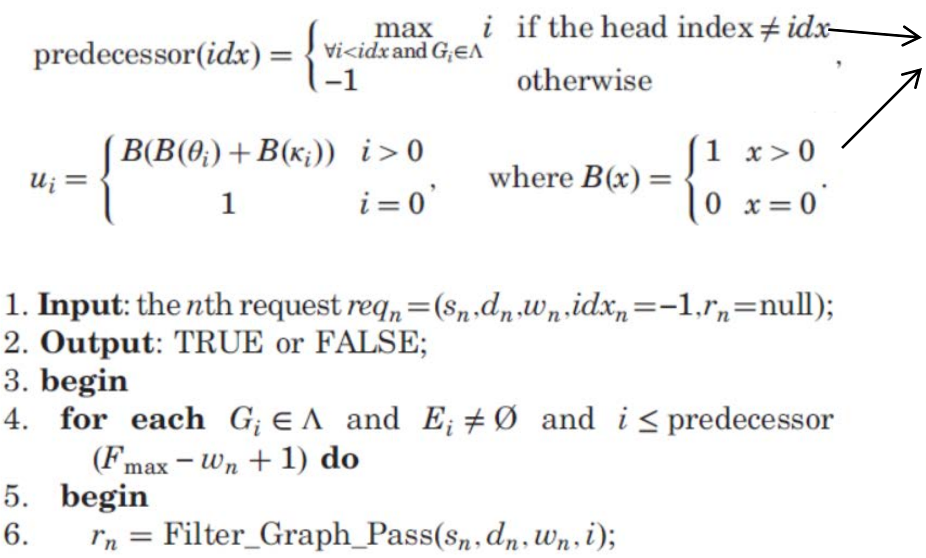
- Filter graph pass
  - the input parameters
    - the source index, the destination index, the demand volume, the starting index
    - Generates a filter graph whose subscript is equal to the last parameter and superscript is i+$W_n - 1$



```
Algorithm 2 Filter_Graph_Pass
Input: $s_n$, $d_n$, $w_n$, $i$; /* source, destination, demand
                                     volume, and starting index */
Output: a routing path $r_n$;
begin
    $r_n \leftarrow$ NULL;
    $end = i + w_n - 1$; /* the superscript of the generated
                             filter graph */
    Generate the filter graph $G_i^{end}$;
    if (there exists a path $p$ in $G_i^{end}$ from $v_{s_n,i}^{end}$ to $v_{d_n,i}^{end}$) then
        Set $r_n$ to the primitive path of $p$;
    return $r_n$;
end
```

Layered Graph Model

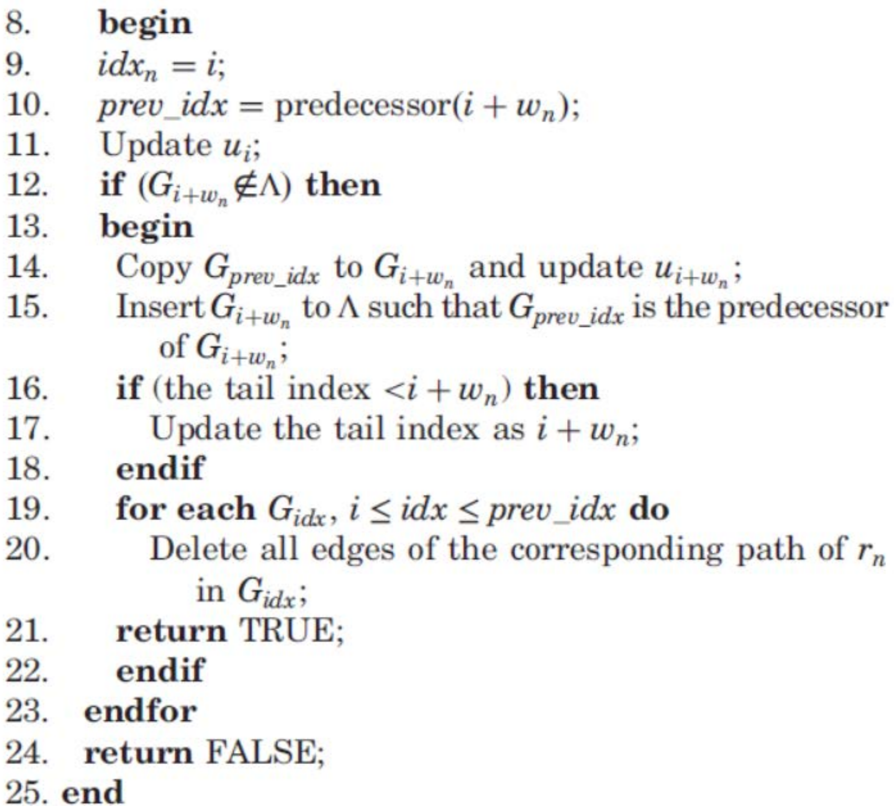
- Layered_Graph_First_Fit
- Update process



$$\text{predecessor}(idx) = \begin{cases} \max\limits_{\forall i<idx \text{ and } G_i\in\Lambda} i & \text{if the head index} \neq idx \\ -1 & \text{otherwise} \end{cases},$$

$$u_i = \begin{cases} B(B(\theta_i)+B(\kappa_i)) & i>0 \\ 1 & i=0 \end{cases}, \quad \text{where } B(x) = \begin{cases} 1 & x>0 \\ 0 & x=0 \end{cases}.$$

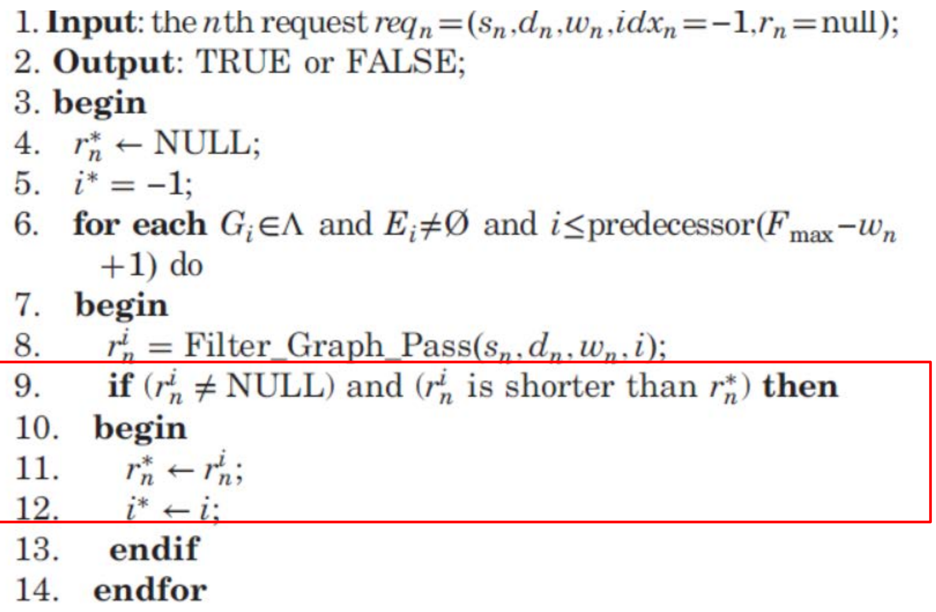
```
Input: the nth request $req_n=(s_n,d_n,w_n,idx_n=-1,r_n=\text{null})$;
Output: TRUE or FALSE;
begin
  for each $G_i \in \Lambda$ and $E_i \neq \emptyset$ and $i \le$ predecessor
      $(F_{\max} - w_n + 1)$ do
  begin
    $r_n$ = Filter_Graph_Pass($s_n, d_n, w_n, i$);
```



```
    begin
    $idx_n = i$;
    $prev\_idx$ = predecessor($i + w_n$);
    Update $u_i$;
    if ($G_{i+w_n} \notin \Lambda$) then
    begin
      Copy $G_{prev\_idx}$ to $G_{i+w_n}$ and update $u_{i+w_n}$;
      Insert $G_{i+w_n}$ to $\Lambda$ such that $G_{prev\_idx}$ is the predecessor
        of $G_{i+w_n}$;
      if (the tail index $< i + w_n$) then
        Update the tail index as $i + w_n$;
      endif
      for each $G_{idx}$, $i \le idx \le prev\_idx$ do
        Delete all edges of the corresponding path of $r_n$
          in $G_{idx}$;
      return TRUE;
      endif
  endfor
  return FALSE;
end
```

Layered Graph Model

- Layered_Graph_Shortest_Path



```
Input: the $n$th request $req_n = (s_n, d_n, w_n, idx_n = -1, r_n = \text{null})$;
Output: TRUE or FALSE;
begin
  $r_n^* \leftarrow$ NULL;
  $i^* = -1$;
  for each $G_i \in \Lambda$ and $E_i \neq \emptyset$ and $i \leq$ predecessor$(F_{\max} - w_n + 1)$ do
  begin
    $r_n^i$ = Filter_Graph_Pass$(s_n, d_n, w_n, i)$;
    if $(r_n^i \neq$ NULL$)$ and $(r_n^i$ is shorter than $r_n^*)$ then
    begin
      $r_n^* \leftarrow r_n^i$;
      $i^* \leftarrow i$;
    endif
  endfor
```

Shortest_Path

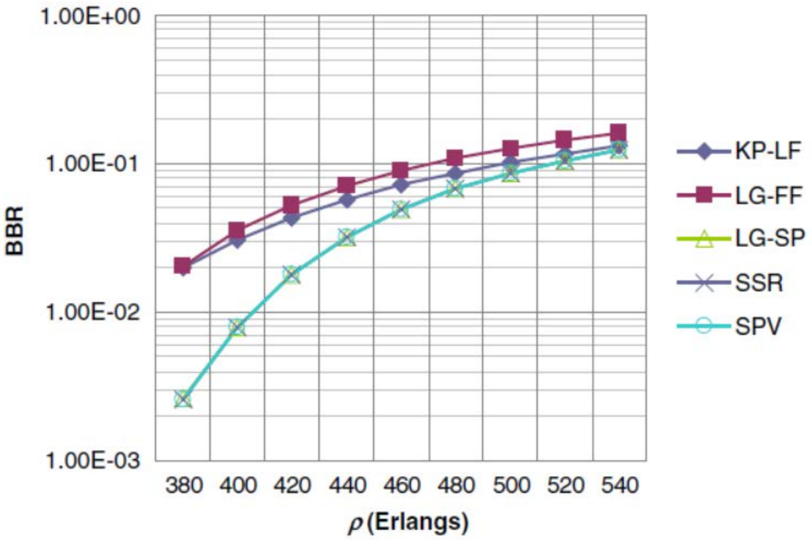
Performance evaluation

- Simulation conditions
  - Arrival traffic requests is a Poisson process
  - Holding time follows exponential distribution
  - Source–destination pairs are randomly generated
  - Assess three RSA algorithms for comparison, including k shortest paths with lowest-index first (KP-LF), SPV[1], and SSR[2]
  - Test Topology: NSFNET, BT
  - Maximum FS number: 380
  - Demand volume is uniformly distributed in the interval [1,15], and the guard band is one slot
  - 10^6 requests are generated for a simulation instance

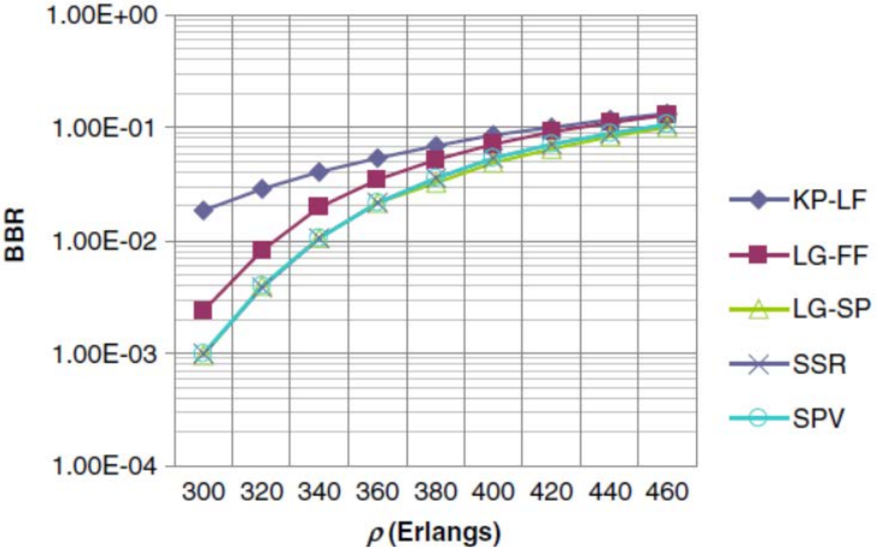
1. Spectrum-Constraint Path Vector Searching Algorithm (SPV)
2. Grid-based spectrum-scan routing (SSR)

Performance evaluation

- BBR



NSFNET

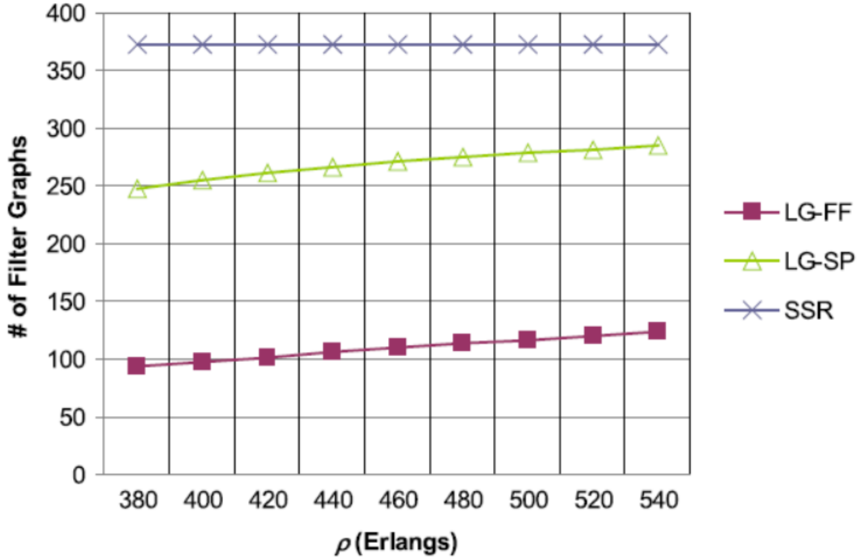


BT

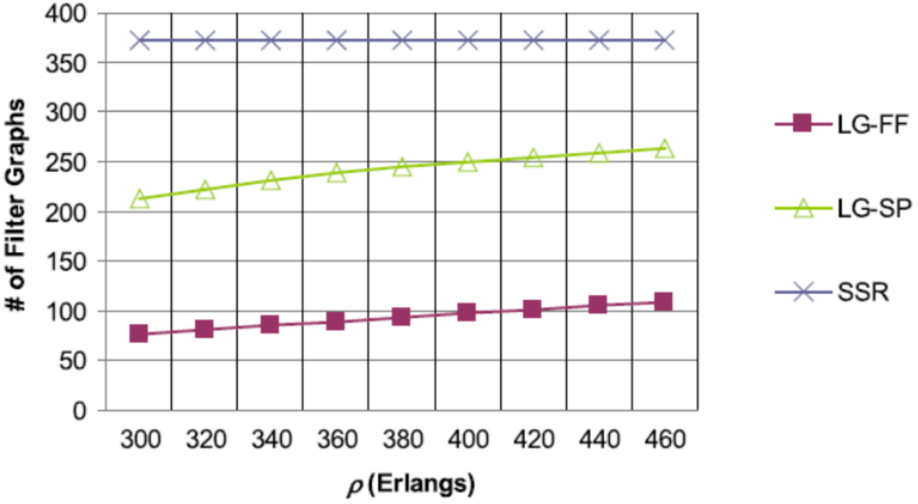
- **LG-SP can achieve almost the same performance level as do SPV and SSR in both topologies**
- **The performance gain of LG-SP over KP-LF ranges from 7% to 87% in NSFNET, and from 23% to 94% in BT**
- **LG-FF is much more prominent than KPLF. However, it performs even worse than KP-LF does in NSFNET**

Performance evaluation

- Number of Generated Filter Graphs



NSFNET

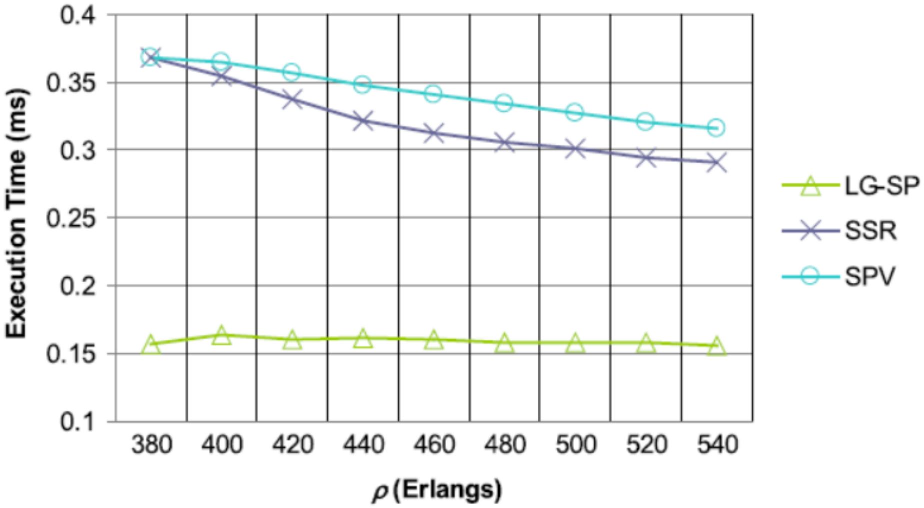


BT

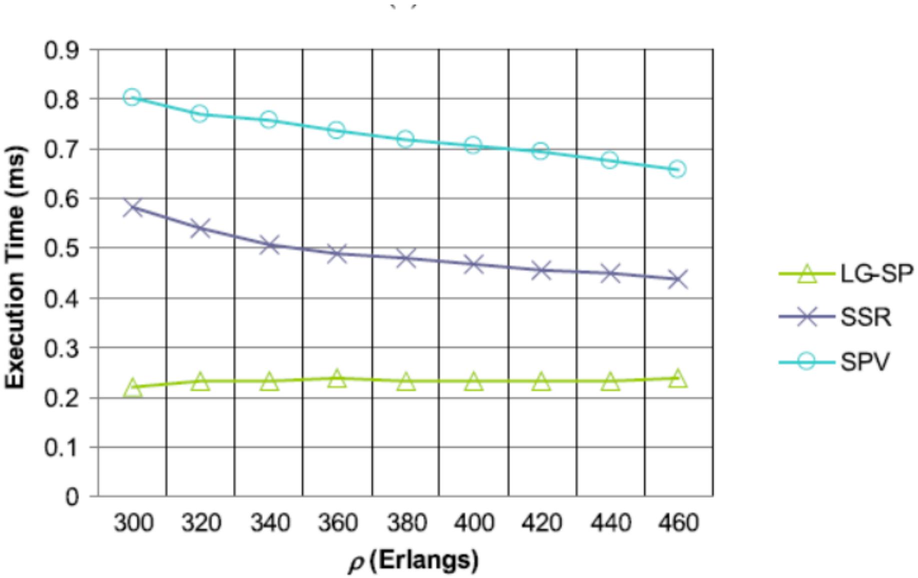
- **LG-SP outperforms SSR by 20% to 34% in NSFNET and by 28% to 44% in BT.**
- **LG-FF and LG-SP both require more filter graphs with increasing traffic load**
- **The slope of LG-SP's curve is slightly higher than that of LG-FF's curve**

Performance evaluation

- Execution Time



NSFNET



BT

- **SPV is the most time-consuming approach among the three algorithms because of its exponential time complexity**
- **it is clear that the execution time of the RSA in BT is about 1.5 times that in NSFNET for LG-SP and SSR, while it is twice for SPV**

Conclusion

- Propose a layered graph model to represent spectrum utilization status and use it to deal with the RSA issue
- Under a dynamic regime, they propose two layered-graph model-based heuristic algorithms, named LG-FF and LG-SP, with different policies to pick out the final RSA decision from candidates
- With a layered graph model, both heuristics can effectively reduce computational complexity to the polynomial level of network scale
- In addition to computational complexity, LG-SP can achieve blocking performance almost as excellent as can SPV

Thank you for your attention!

Presenter: Yongcheng(jeremy) Li

PhD student, School of Electronic and Information Engineering,
Soochow University, China
Email: liyongcheng621@163.com

Group Meeting, Friday, July 7, 2016