

# زنجیره‌سازی کارکردهای مجازی سرویس شبکه با در نظر گرفتن محدودیت منابع مدیریتی

---

پرهام الوانی

شهریور ۱۳۹۸

دانشکده مهندسی کامپیوتر

دکتر بهادر بخشی



دانشکده مهندسی  
کامپیوتر و فناوری اطلاعات

## فهرست

۱ مقدمه

۲ سابقه‌ی کارها

۳ تعریف مساله

۴ فرمول‌بندی و مدل‌سازی ریاضی مساله

۵ راه‌حل پیشنهادی

۶ ارزیابی

## ۱. مقدمه

---

## شبکه‌های سنتی

- ◀ یک سرویس شبکه به صورت تعدادی کارکرد مشخص که ترافیک با ترتیب مشخصی از آن ها عبور می‌کند، تعریف می‌شود.
- ◀ کارکردهای شبکه به صورت سخت‌افزار و نرم‌افزار اختصاصی تهیه شده از سازندگان مختلف استفاده می‌شوند.
- ◀ کارکردها باید در مکان مناسب در شبکه قرار گیرند و ترافیک به سمت آن‌ها هدایت شود.

## شبکه‌های سنتی

◀ افزایش نیازمندی به سرویس‌های متنوع با عمر کوتاه و نرخ بالای ترافیک

- خریداری، انبارداری و استقرار سخت‌افزارهای اختصاصی
- افزایش هزینه‌های خرید، آموزش و انبارداری
- کاهش فضای فیزیکی
- سرشار آموزش کارکنان
- محدودیت نوآوری در سخت‌افزار و سرویس

## Network Functions Virtualization

مجازی‌سازی کارکردهای شبکه

## شبکه‌های سنتی

- ◀ ترافیک کاربر باید از تعدادی کارکرد شبکه به ترتیب معینی عبور کند.
- ◀ کارکردها به صورت سخت‌افزاری به یکدیگر متصلند و ترافیک با استفاده از جداول مسیریابی به سمت آن‌ها هدایت می‌شود.
- ◀ نیاز به تغییر همبندی سریع و یا مکان کارکردها برای سرویس‌دهی بهتر
  - استقرار و تغییر ترتیب کارکردها دشوار است
  - امکان رخدادن خطاهای متعدد

## Service Function Chaining

زنجیره‌سازی کارکرد سرویس

## معماری پیشنهادی

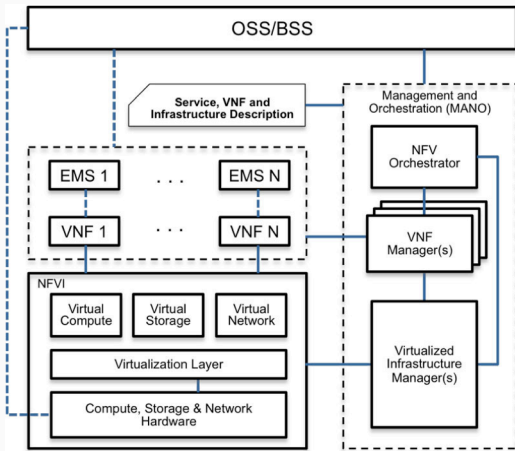
### ◀ مجازی‌سازی کارکردهای شبکه

- اواخر سال ۲۰۱۲، ETSI NFV ISG توسط هفت اپراتور جهانی شبکه تأسیس شد.
- اکنون بیش از ۲۵۰ سازمان با آن همکاری می‌کنند.
- اجرای کارکردها بر روی سرورهای استاندارد با توان بالا به وسیله مجازی‌سازی کارکردها
- کاهش نیاز به تجهیزات سخت‌افزاری خاص منظوره
- اشتراک گذاری منابع بین کارکردها
- کاهش هزینه‌های تجهیزات و مصرف انرژی از طریق تجمیع کارکردها

### ◀ زنجیره‌سازی کارکرد سرویس

- امکان تعریف زنجیره کارکردها به صورت پویا و بدون تغییر در زیرساخت فیزیکی
- RFC 7665

# معماری پیشنهادی



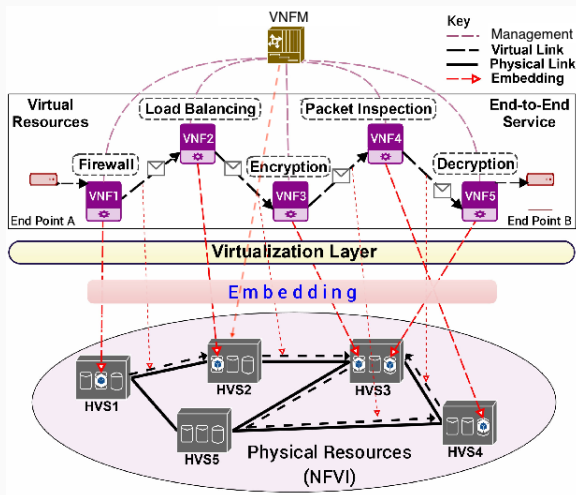
شکل ۱: معماری سطح بالای مجازی‌سازی کارکردهای شبکه



## مقدمه

- ◀ NFVO وظیفه‌ی استقرار زنجیره‌های کارکرد سرویس را برعهده دارد.
- ◀ VNFM مسئول چرخه‌ی زندگی کارکردهای مجازی شبکه می‌باشد.
- ◀ چرخه‌ی زندگی هر کارکرد مجازی شامل عملیات‌هایی همچون نمونه‌سازی، مقیاس‌کردن، به‌روزرسانی و پایان دادن می‌باشد.
- ◀ هر نمونه از کارکردهای مجازی شبکه نیاز دارد تحت مدیریت یکی از VNFM‌های موجود در شبکه باشد.

## چالش‌ها



◀ مدیریت و هماهنگی

◀ مصرف بهینه‌ی انرژی

◀ تخصیص منابع به

کارکردهای مجازی

◀ مسیریابی زنجیره‌های

کارکرد سرویس

◀ پذیرش زنجیره‌های

کارکرد سرویس

◀ به روزرسانی و مقیاس

کردن کارکردهای مجازی

سرویس

## ۲. سابقه‌ی کارها

---

## سابقه‌ی کارها

## جدول ۱: مقایسه مقالات پذیرش زنجیره‌های کارکرد سرویس

منبع	منابع تخصیص یافته	محدودیت ظرفیت پردازشی نمونه	برخط یا برون خط	نگاشت کارکرد و لینک	انتساب کارکرد	اشتراک نمونه	تخصیص VNFM									
#	other	MEM	BW	CPU	دارد	ندارد	برخط	برون خط	کارکرد	لینک	یک نمونه	چند نمونه	دارد	ندارد	دارد	ندارد
[۳]	—	—	✓	✓	—	✓	—	✓	✓	✓	✓	—	—	✓	—	✓
[۴]	—	—	✓	✓	✓	—	✓	—	✓	✓	—	✓	—	✓	—	✓
[۵]	—	—	✓	✓	✓	—	✓	—	✓	✓	—	✓	—	✓	—	✓
[۱]	—VNFM capacity	—	—	—	✓	—	✓	—	—	✓	—	—	—	✓	—	—
پژوهش حاضر	✓	✓	✓	✓	—	—	✓	—	✓	✓	✓	—	—	✓	✓	—

## سابقه‌ی کارها

- ◀ این مقاله مساله‌ی جایگذاری VNFMs را مطرح می‌کند.
- ◀ این مقاله فرض می‌کند زنجیره‌های جایگذاری شده‌اند و هر در بازه‌ی زمانی می‌توانند بازنگاشت شوند.
- ◀ این مساله قصد دارد با در نظر گرفتن هزینه‌های عملیاتی مساله‌ی بازنگاشت VNFMs در بازه‌های زمانی را حل کند.

Mohammad Abu-Lebdeh et al. “On the Placement of VNF Managers in Large-Scale and Distributed NFV Systems”. In: *IEEE Transactions on Network and Service Management* 14.4 (Dec. 2017), pp. 875–889. DOI: 10.1109/tnsm.2017.2730199. URL: <https://doi.org/10.1109/tnsm.2017.2730199>

### ۳. تعریف مساله

---

## تعریف مساله

بیشینه‌سازی سود حاصل از پذیرش زنجیره‌های کارکرد سرویس با در نظر گرفتن کارکرد سرویس با در نظر گرفتن نیاز برخی از نمونه‌های کارکرد مجازی شبکه به VNFM.

## تعریف مساله

- ◀ توپولوژی زیرساخت شامل پنهای باند لینک‌ها و ظرفیت NFVI-PoP‌ها، موجود است.
- ◀  $n$  تقاضای زنجیره کارکرد سرویس به صورت کامل و از پیش مشخص شده داریم.
- ◀ هر تقاضا شامل نوع و تعداد نمونه‌های مجازی، پنهای باند لینک‌های مجازی و توپولوژی نمونه‌های مجازی می‌باشد.



## تعریف مساله

- ◀ نمونه‌ها بین زنجیره‌ها به اشتراک گذاشته نمی‌شوند.
- ◀ محدودیت ظرفیت لینک‌ها
- ◀ محدودیت توان پردازش سرورهای فیزیکی با توجه به میزان حافظه و تعداد پردازنده‌ها
- ◀ برخی از سرورهای فیزیکی نمی‌توانند سرورهای فیزیکی مشخصی را مدیریت کنند.
- ◀ برخی از سرورهای فیزیکی توانایی پشتیبانی از کارکردهای مجازی را ندارد.
- ◀ برخی از نمونه‌های کارکرد مجازی تنها می‌توانند روی سرورهایی خاص نگاشته شوند.

## تعریف مساله

- ◀ برای مدیریت یک‌دست و آسان‌تر زنجیره‌ها و در عین حال جمع‌آوری راحت‌تر خطاها، برای هر زنجیره یک VNFM تخصیص می‌دهیم.
- ◀ VNFM ها می‌توانند بین زنجیره به اشتراک گذاشته شوند.
- ◀ هر نمونه از VNFM ها می‌تواند تعداد مشخصی از نمونه‌های کارکرد مجازی شبکه را سرویس دهد.
- ◀ برای ارتباط میان هر نمونه از VNFM ها و VNF ها پهنای باند مشخصی رزرو می‌گردد.
- ◀ در صورتی که NFVI-PoP بتواند از VNFM پشتیبانی نماید، می‌توان به هر تعداد که ظرفیت آن اجازه می‌دهد بر روی آن VNFM نصب نمود.

## چالش‌ها و نوآوری‌های مساله

- ◀ در نظر گرفتن نیازمندی نمونه‌های کارکرد مجازی به یک VNFM
- ◀ در نظر گرفتن نیازمندی تاخیر برای لینک‌های مدیریتی
- ◀ تخصیص منابع مدیریتی به زنجیره‌ها و مسیریابی ارتباط مدیریتی
- ◀ جایگذاری و مسیریابی توانان زنجیره‌های کارکرد سرویس
- ◀ طراحی مساله‌ی نزدیک به واقعیت

## معیار و نحوه‌ی ارزیابی

- ◀ مدل‌سازی مساله
- ◀ حل مساله‌ی بهینه در ابعاد کوچک
- ◀ پیاده‌سازی راه‌حل مکاشفه‌ای
- ◀ معیار مقایسه این راه حل سود حاصل از پذیرش تقاضاهای زنجیره‌های کارکرد سرویس می‌باشد.
- ◀ مقایسه‌ی نتایج راه‌حل مکاشفه‌ای با جواب بهینه

## ۴. فرمول‌بندی و مدل‌سازی ریاضی مساله

---

## فرمول‌بندی

هدف اصلی مساله پذیرش بیشترین تعداد تقاضا می‌باشد. در اینجا فرض می‌کنیم پذیرش هر تقاضا سودی منحصر به فرد و هزینه‌ای برای تهیه گواهی VNFم در بر خواهد داشت. بنابراین تابع هدف به شکل زیر می‌باشد:

$$\max \sum_{h=1}^T c_h x_h - \sum_{w \in V_s^{PN}} licenseFee.w \bar{y}_w \quad (1)$$

## فرمول‌بندی

محدودیت حافظه نودها

$$\sum_{k=1}^F y_{wk} memory(k) + \bar{y}_w me\bar{m}ory \leq N_{ram}^{PN}(w) \quad \forall w \in V_s^{PN} \quad (2)$$

محدودیت تعداد پردازنده‌های نودها

$$\sum_{k=1}^F y_{wk} core(k) + \bar{y}_w c\bar{o}re \leq N_{core}^{PN}(w) \quad \forall w \in V_s^{PN} \quad (3)$$

## فرمول‌بندی

اگر تقاضای  $h$ ام پذیرفته شده باشد می‌بایست تمام VNF node های آن سرویس شده باشند.  
یک VNF حداکثر یکبار سرویس داده شود.

$$x_h = \sum_{k=1}^F \sum_{w \in V_s^{PN}} z_{vw}^k \quad \forall v \in V_{h,F}^{SFC}, \forall h \in [1, \dots, T] \quad (4)$$



## فرمول‌بندی

اگر تقاضای  $h$  پذیرفته شده باشد می‌بایست توسط یک VNFM سرویس شده باشد. توجه شود که این محدودیت اجازه‌ی تخصیص بیش از یک VNFM به زنجیره نمی‌دهد.

$$x_h = \sum_{w \in V_s^{PN}} \bar{z}_{hw} \quad \forall h \in [1, \dots, T] \quad (5)$$

## فرمول‌بندی

محدودیت ظرفیت سرویس‌دهی VNFM این محدودیت براساس تعداد ماشین‌های مجازی که هر VNFM سرویس می‌دهد تعیین شده است. در نظر داشته باشید که ممکن است برخی از انواع VNF ها نیازی به مدیریت شدن نداشته باشند.

$$\sum_{i=1}^T \bar{z}_{iw} \cdot (\text{len}(i) - \sum_{v \in V_{i,F}^{SFC}} \sum_{k \in [1, \dots, F]} \text{type}(v, k) \cdot \text{isManageable}(k)) \leq \text{capacity} \cdot \bar{y}_w \quad \forall w \in V_s^{PN} \quad (6)$$

## فرمول‌بندی

اگر  $VNF$ ،  $v$  توسط نمونه‌ای نوع  $k$  روی سرور  $w$  سرویس می‌شود می‌بایست این  $VNF$  از نوع  $k$  ام باشد.

$$z_{vw}^k \leq type(v, k) \quad \forall w \in V_s^{PN}, \forall k \in [1, \dots, F], \forall v \in \cup_{i=1}^T V_{i,F}^{SFC} \quad (7)$$

## فرمول‌بندی

در صورتی که سرور  $w$  توانایی اجرای نمونه‌های VNF را نداشته باشد نباید نمونه‌ای روی آن قرار گیرد.

$$\sum_{k \in [1, \dots, F]} y_{wk} \leq M.vnfSupport(w) \quad w \in V_s^{PN} \quad (8)$$

## فرمول‌بندی

برخی از سرورهای نمی‌توانند توسط سرورهای مشخصی مدیریت شوند. این ویژگی به ادمین شبکه امکان مدیریت بیشتری می‌دهد و او می‌تواند با دست باز تمامی سیاست‌های مورد نظرش را اعمال نماید.

$$\begin{aligned}
 1 - z_{vw_1}^k + \bar{z}_{hw_2} &= 0 \quad \forall w_1 \in V_s^{PN} \forall w_2 \in V_s^{PN} \\
 notManagableBy(w_1, w_2) &= 1 \\
 \forall h \in [1, \dots, T], \forall v \in V_{h,F}^{SFC}, \forall k \in [1, \dots, T]
 \end{aligned}
 \tag{9}$$

## Flow Conservation

$$\sum_{(i,j) \in E^{PN}} \tau_{ij}^{(u,v)} - \sum_{(j,i) \in E^{PN}} \tau_{ji}^{(u,v)} = \sum_{k=1}^F z_{ui}^k - \sum_{k=1}^F z_{vi}^k$$

$$\forall i \in V_S^{PN}, (u, v) \in E_h^{SFC}, h \in [1, \dots, T] \quad (10)$$

## فرمول‌بندی

## Flow Conservation

$$\sum_{(i,j) \in E^{PN}} \bar{\tau}_{ij}^v - \sum_{(j,i) \in E^{PN}} \bar{\tau}_{ji}^v = \sum_{k=1}^F z_{vi}^k - \bar{z}_{hi} \quad \forall i \in V_S^{PN}, v \in V_{h,F}^{SFC}, h \in [1, \dots, T] \quad (11)$$

## فرمول‌بندی

محدودیت ظرفیت لینک‌ها

$$\sum_{v \in \cup_{i=1}^T V_{i,F}^{SFC}} \bar{\tau}_{ij}^v * bandwidth + \sum_{(u,v) \in \cup_{i=1}^T E_i^{SFC}} \tau_{ij}^{(u,v)} * bandwidth(u,v) \leq C_{ij}$$

$$\forall (i,j) \in E^{PN}$$

$$(12)$$



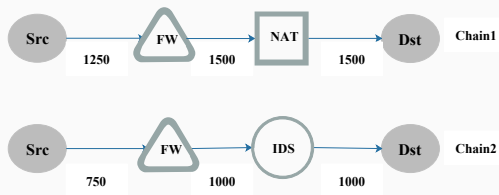
## فرمول‌بندی

شعاع همسایگی تضمین می‌کند که زمان سرویس‌دهی توسط VNFMs ها در یک بازه مشخص (از نظر تعداد گام) خواهد بود.

$$\sum_{(i,j) \in E^{PN}} \bar{\tau}_{ij}^v \leq radius \quad \forall v \in \cup_{i=1}^T V_{i,F}^{SFC} \quad (13)$$

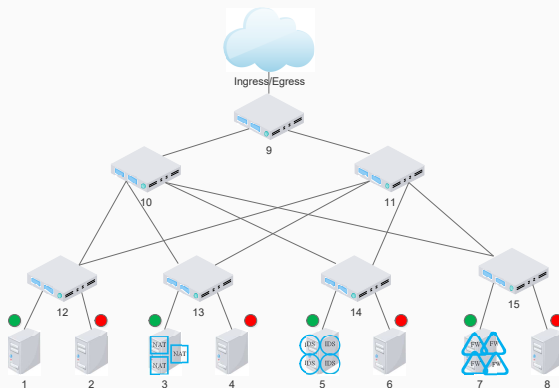
## مساله‌ی نمونه

زنجیره‌های زیر را به عنوان تقاضاها در نظر می‌گیریم.



## مساله‌ی نمونه

فرض می‌کنیم مرکز داده‌ای دارای توپولوژی زیر می‌باشد.



## مساله‌ی نمونه

جدول ۲: نیازمندی نمونه‌های مساله‌ی نمونه

Spec/VNF	vFW	vNAT	vIDS
CPU (vCore)	2	2	2
Memory (GB)	2	4	2

## مساله‌ی نمونه

شکل ۲: مشخصات سرورهای زیرساخت مساله‌ی نمونه

	Server 1,2,7,8	Servers 3,4,5,6
Installed vCPU	144	72
Installed Memory (GB)	1408	288
Link (Gbps)	40	40

## مساله‌ی نمونه

- ◀ نمونه‌ها تنها می‌توانند روی سرورهای ۱، ۳، ۵ و ۷ قرار گیرند.
- ◀ مدیریت سرورهای ۱ و ۳ تنها می‌تواند روی سرورهای ۲ و ۴ صورت گیرد،
- ◀ مدیریت سرور ۵ تنها می‌تواند روی سرورهای ۴ و ۶ صورت گیرد.
- ◀ مدیریت سرور ۷ تنها می‌تواند روی سرورهای ۶ و ۸ صورت گیرد.

## مساله‌ی نمونه

- ◀ هر VNFM تنها می‌تواند ۵ نمونه را پشتیبانی کند.
- ◀ هر VNFM نیاز به ۴ گیگابایت حافظه و ۲ هسته‌ی پردازشی دارد.

## مساله‌ی نمونه

جدول ۳: نتایج مساله‌ی نمونه

	Src	Node-0	Node-1	Dst	VNFM
Chain 0	Switch-9	Server-7	Server-5	Switch-9	Server-6
Chain 1	Switch-9	Server-3	Server-3	Switch-9	Server-4



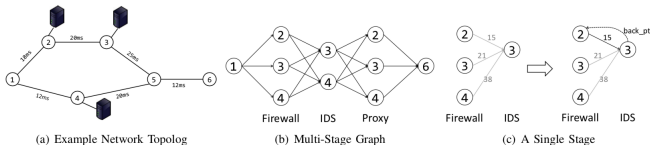
## ۵. راه‌حل پیشنهادی

---

## راه‌حل پیشنهادی

- ◀ مساله‌ی اصلی یک مساله‌ی NP-Hard می‌باشد.
- ◀ برای حل مساله در زمان معقول برای ابعاد بزرگ نیاز به یک الگوریتم مکاشفه‌ای می‌باشد.
- ◀ از ایده‌ی الگوریتم [2] برای جایگذاری زنجیره‌ها شروع می‌کنیم.

## ایده‌ی اصلی



◀ الگوریتم برای جایگذاری زنجیره از یک گراف چند مرحله‌ای استفاده می‌کند.

◀ در هر مرحله جایگذاری مرحله‌ی قبلی نهایی می‌شود و بر اساس آن یک مجموعه‌ی امکان‌پذیر شکل می‌گیرد.

## Joint Service Deployment - Manager Placement ◀

- ◀ زنجیره‌ها را با استفاده از الگوریتم [2] جایگذاری می‌کنیم.
- ◀ در زمان انتخاب مجموعه‌ی امکان‌پذیر محدودیت‌های مساله را اعمال می‌کنیم.
- ◀ بعد از جایگذاری هر زنجیره VNFم آن را انتخاب می‌کنیم.
- ◀ برای انتخاب VNFم اولویت با نمونه‌هایی است که ظرفیت آن‌ها کامل استفاده نشده است.
- ◀ در بین VNFم‌هایی که ظرفیت خالی دارند اولویت با نمونه‌هایی است که منابع پردازشی بیشتری دارند.

## eJSD-MP

- ◀ الگوریتم پیشنهادی JSD-MP زمان اجرای زیادی دارد که می‌توان آن را کاهش داد.
- ◀ الگوریتم پیشنهادی eJSD-MP از برون‌خط بودن مساله استفاده نمی‌کند.
- ◀ برای استفاده از ویژگی برون‌خط بودن مساله زنجیره‌ها را بر اساس قیمت‌شان مرتب می‌کنیم.
- ◀ برای کاهش زمان اجرای الگوریتم نسب مشخصی از زنجیره‌ها را با الگوریتم first-fit جایگذاری می‌کنیم.
- ◀ enhanced JSD-MP

## ۶. ارزیابی

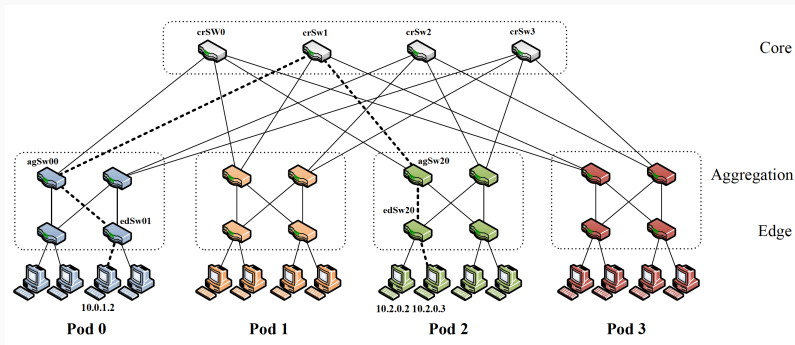
---

## پیاده‌سازی بهینه

فرمول‌بندی ارائه شده بر روی نرم‌افزار CPLEX که محصول شرکت IBM بوده و برای حل مسائل برنامه‌ریزی خطی و ... استفاده می‌شود، به زبان جاوا پیاده‌سازی شده است.

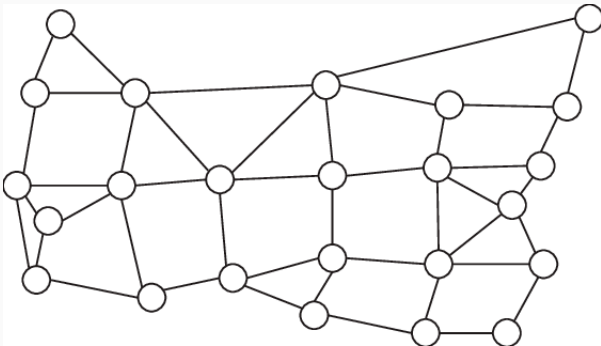


# توپولوژی FatTree





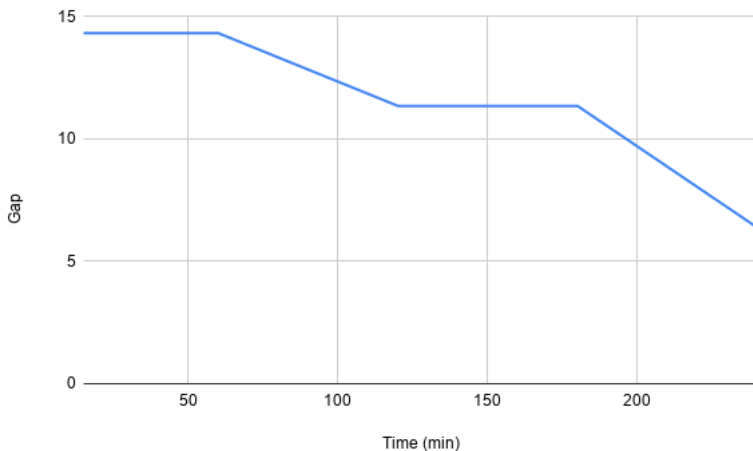
## توپولوژی USnet



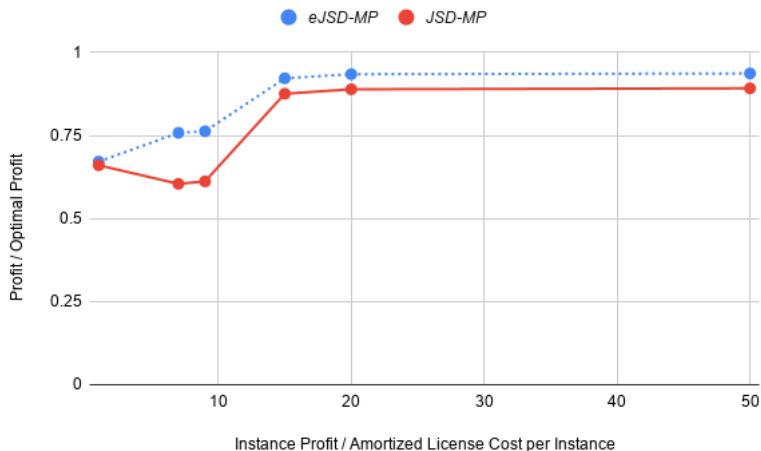
## محیط ارزیابی

- ◀ برای ارزیابی از زنجیره‌های تصادفی استفاده می‌شود و هر نمونه از ارزیابی میانگین ۱۰ اجرا می‌باشد.
- ◀ زنجیره‌های تولید شده دارای گره‌ی آغازی و پایانی می‌باشند و ترافیک عبوری از آن‌ها ۲۵۰ واحد است.

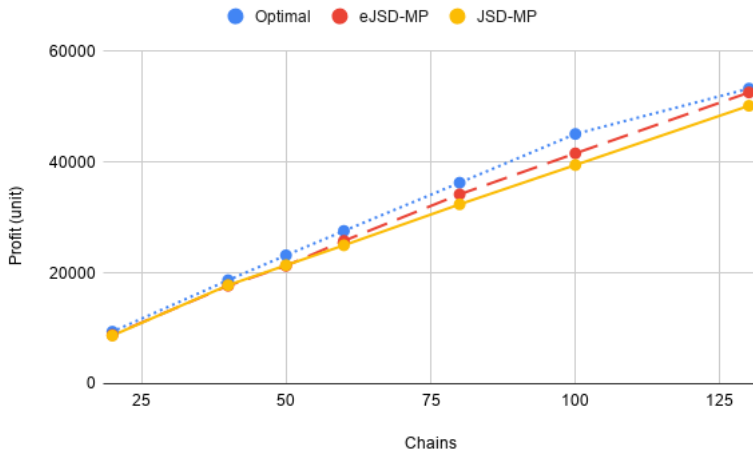
## شکاف بهینه الگوریتم بهینه



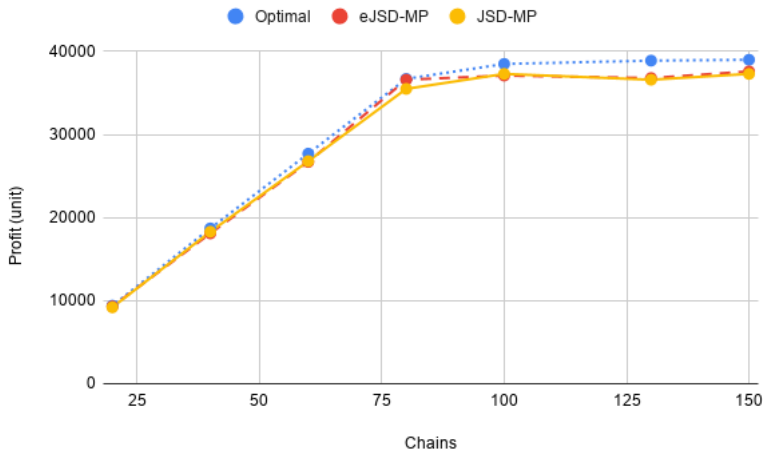
## نسبت سود به هزینه



# FatTree سود نهایی در توپولوژی





# USnet سود نهایی در توپولوژی



## جمع‌بندی

- ◀ هر دو الگوریتم ارائه شده سود نهایی نزدیکی به الگوریتم بهینه ارائه می‌کنند.
- ◀ الگوریتم eJSD-MP در زمان کمتر نتایجی بهتر یا برابر با الگوریتم JSD-MP ارائه می‌کند.

-  Mohammad Abu-Lebdeh et al. “On the Placement of VNF Managers in Large-Scale and Distributed NFV Systems”. In: *IEEE Transactions on Network and Service Management* 14.4 (Dec. 2017), pp. 875–889. DOI: 10.1109/tnsm.2017.2730199. URL: <https://doi.org/10.1109/tnsm.2017.2730199>.
-  Md. Faizul Bari et al. “On orchestrating virtual network functions”. In: *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE, Nov. 2015. DOI: 10.1109/cnsm.2015.7367338. URL: <https://doi.org/10.1109/cnsm.2015.7367338>.
-  V. Eramo, A. Tosti, and E. Miucci. “Server Resource Dimensioning and Routing of Service Function Chain in NFV Network Architectures”. In: *Journal of Electrical and Computer Engineering* 2016 (2016), pp. 1–12. DOI: 10.1155/2016/7139852. URL: <https://doi.org/10.1155/2016/7139852>.
-  Milad Ghaznavi et al. “Distributed Service Function Chaining”. In: *IEEE Journal on Selected Areas in Communications* 35.11 (Nov. 2017), pp. 2479–2489. DOI: 10.1109/jsac.2017.2760178. URL: <https://doi.org/10.1109/jsac.2017.2760178>.
-  Huawei Huang et al. “Near-Optimal Deployment of Service Chains by Exploiting Correlations between Network Functions”. In: *IEEE Transactions on Cloud Computing* (2017), pp. 1–1. DOI: 10.1109/tcc.2017.2780165. URL: <https://doi.org/10.1109/tcc.2017.2780165>.



## فرمول‌بندی

## پارامترهای مساله

$memory(k)$	required RAM of VNF instance with type $k$ in GB
$core(k)$	required CPU cores of VNF instance with type $k$
$memôry$	required RAM of VNFM in GB
$côre$	required CPU cores of VNFM
$capacity$	maximum number of VNF instances that VNFM can handle
$len(h)$	number of VNF instances in $h$ th SFC request

## فرمول‌بندی

## پارامترهای مساله

$type(v, k)$	assuming the value 1 if the VNF instance $v$ has type $k$
$bandwidth(u, v)$	required bandwidth in link from VNF instance $u$ to $v$
$\hat{bandwidth}$	required bandwidth in management link
$radius$	maximum neighborhood distance for instance management

## فرمول‌بندی

## پارامترهای مساله

$licenseFee$	VNFM license fee that must pay for each VNFM
$vnfSupport(w)$	assuming the value 1 if the physical server $w$ can support VNF instances
$isManageable(k)$	assuming the value 1 if the type $k$ needs a manager
$notManagableBy(w1, w2)$	assuming the value 1 if the physical server $w1$ cannot manage by physical server $w2$

## فرمول‌بندی

## متغیرهای تصمیم‌گیری

- $x_h$  binary variable assuming the value 1 if the  $h$ th SFC request is accepted; otherwise its value is zero
- $y_{wk}$  the number of VNF instances of type  $k$  that are used in server  $w \in V_s^{PN}$
- $z_{vw}^k$  binary variable assuming the value 1 if the VNF node  $v \in \cup_{i=1}^T V_{i,F}^{SFC}$  is served by the VNF instance of type  $k$  in the server  $w \in V_s^{PN}$

## فرمول‌بندی

متغیرهای تصمیم‌گیری

- $\bar{y}_w$  the number of VNFMs that are used in server  $w \in V_s^{PN}$
- $\bar{z}_{hw}$  binary variable assuming the value 1 if  $h$ th SFC is assigned to VNFM on server  $w \in V_s^{PN}$

## فرمول‌بندی

## متغیرهای تصمیم‌گیری

- $\tau_{ij}^{(u,v)}$  binary variable assuming the value 1 if the virtual link  $(u, v)$  is routed on the physical network link  $(i, j)$
- $\bar{\tau}_{ij}^v$  binary variable assuming the value 1 if the management traffic of VNF node  $v$  is routed on the physical network link  $(i, j)$