Security Function Placement in Cloud Data Centers

Abeer Ali, Christos Anagnostopoulos, Dimitrios P. Pezaros

University of Glasgow, UK

Abstract

Providing security services in multitenant Cloud Data Centers can offer the customised services of on-premises security and deliver the efficiency and high performance of Cloud-based services. We formulate the problem of allocating the security functions over a Cloud DC, and propose a modified version of Best-Fit Decreasing (BFD) heuristic algorithm to solve it. We demonstrate how it can reduce computing resources consumption compared to other heuristic algorithms.

Introduction

As services over the Cloud are offered in a scalable, elastic, and always-on manner, they are extremely prone to security vulnerabilities that results in financial losses due to (e.g. performance degradation) [1]. Furthermore, attacks affect non-targeted tenants and increase the overall energy consumption [2]. While security solution usually deployed in the form of bespoke middleboxes such as, e.g., firewalls and IDS, they are expensive, high-performance and vendor-specific appliances which results in deployment inflexibility, limit extensible functionality and vendor lock-in.

Methodology

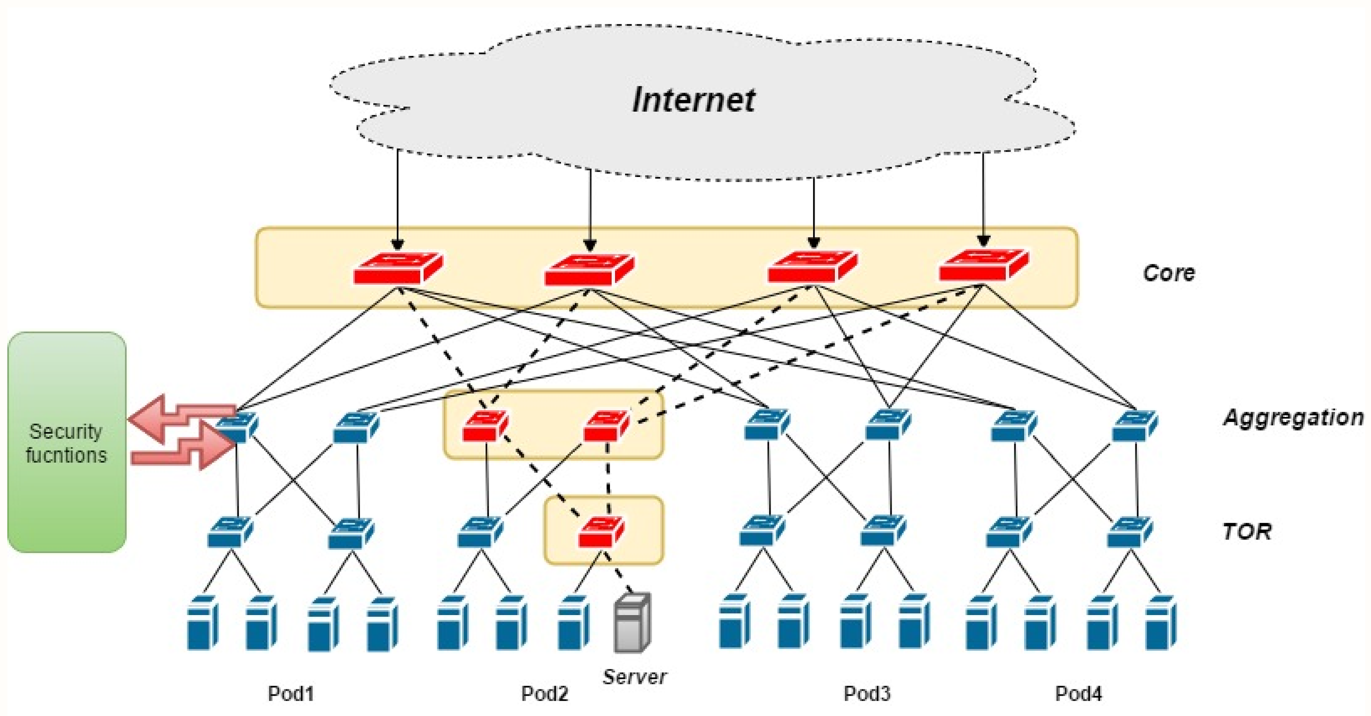

Figure 1: Security services' deployment locations in Cloud DC

We propose a flexible and scalable framework to orchestrate security services in Cloud DCs. It is responsible for allocate, deploy, manage and monitor the security modules. The NFV and SDN technologies are exploited to implement the framework.

The placement problem

The placement problem is satisfying tenants' requests for security services by allocating security modules to the framework deployment locations in the DC infrastructure. The placement is constrained by the limited capacity of locations and must guarantee that the required traffic is passed to the modules.

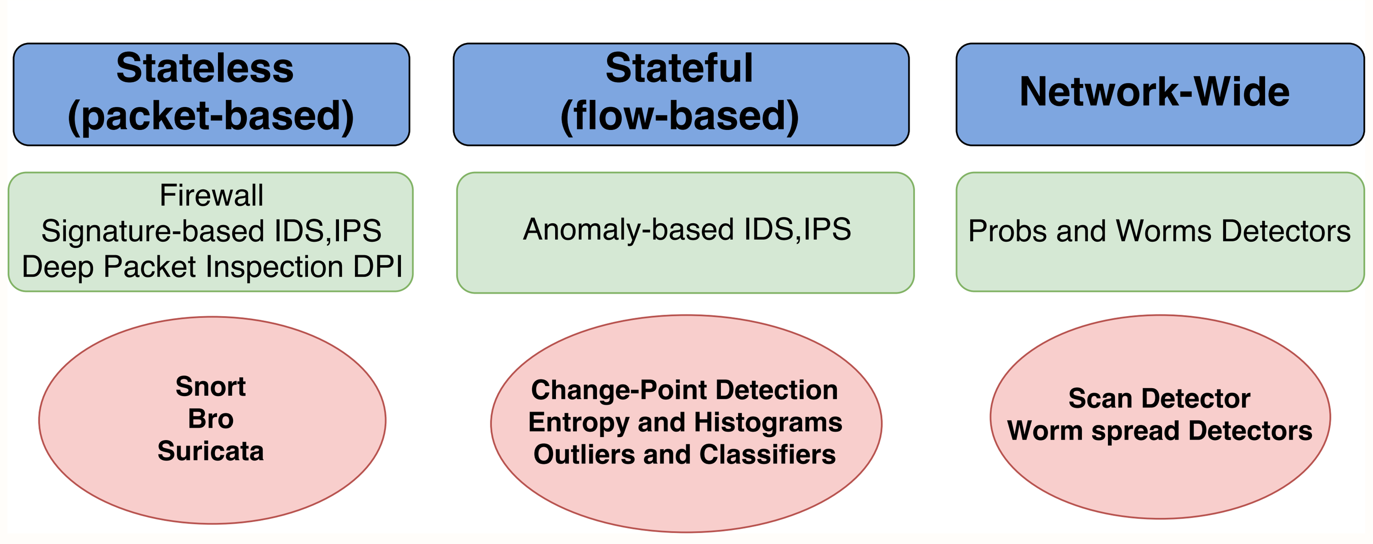

Figure 2: Security equivalence classes based on traffic granularity.

Allocating the security functions of the stateless class is a variable costs variable size bin packing problem (VSBPP) [3] and our objective function is to maximising the residual resources after allocation.

$$Max[\sum_{\forall s\in S} R(s) - \sum_{\forall s\in S}\sum_{\forall (h,m)\in A(s)} R(h,m,s)]$$

s.t.

$$\left(\sum_{\forall (h,m)\in A(s)} R(h,m,s)\right) \leq R(s) \quad \forall s \in S$$

$$Min\left(\sum_{\forall h\in H}\sum_{\forall m\in Q(h)} R(m) \quad if A(h,m) = -1\right)$$

Where an allocation A is represented by $A(h,m)=\begin{cases} l, & \text{Allocated to } l \\ -1 & \text{Otherwise} \end{cases}$ $\forall h \in H, \forall m \in Q(h)$, $R(s)$ is the computing resources capacity available at location s, $R(h,m,s)$ is the function to compute the requested resources of module m to be allocated in location s to protect tenant h, $A(s)$ is defined as the set of requested tuples allocated to switch s.

Modified BFD Algorithm

In modified Best-fit Decreasing (BFD), requests are sorted in decreasing order by required resources then selecting allocations that cause the least increase in computing resources.

Input: H, M, S, Q, L

Output: A, Unsatisfied_Requests

```
Sorted_Q ← sort(Q) // sort by requested resources
for all h, m ∈ Sorted_Q do
  min_cost ← MAX
  level_found ← −1
  for all l ∈ L do
    if (A.Enough_Resources(h, m, l) == TRUE) then
      cost ← calculate_cost(A, h, m, l)
      if (cost < min_cost) then
        level_found ← l
        min_cost ← cost
      end if
    end if
    if (level_found ≠ −1) then
      A ← A.Allocate(h, m, l)
    else
      Unsatisfied_Requests ← h, m
    end if
  end for
end for
return A, Unsatisfied_Requests
```

Results

Evaluation of BFD, First-Fit deceasing (FFD), Best-fit (BF) and First-fit (FF) heuristic algorithms for the placement problem under uniform distribution for module requests in Fat-tree DC size=4 is shown in Figure 3.

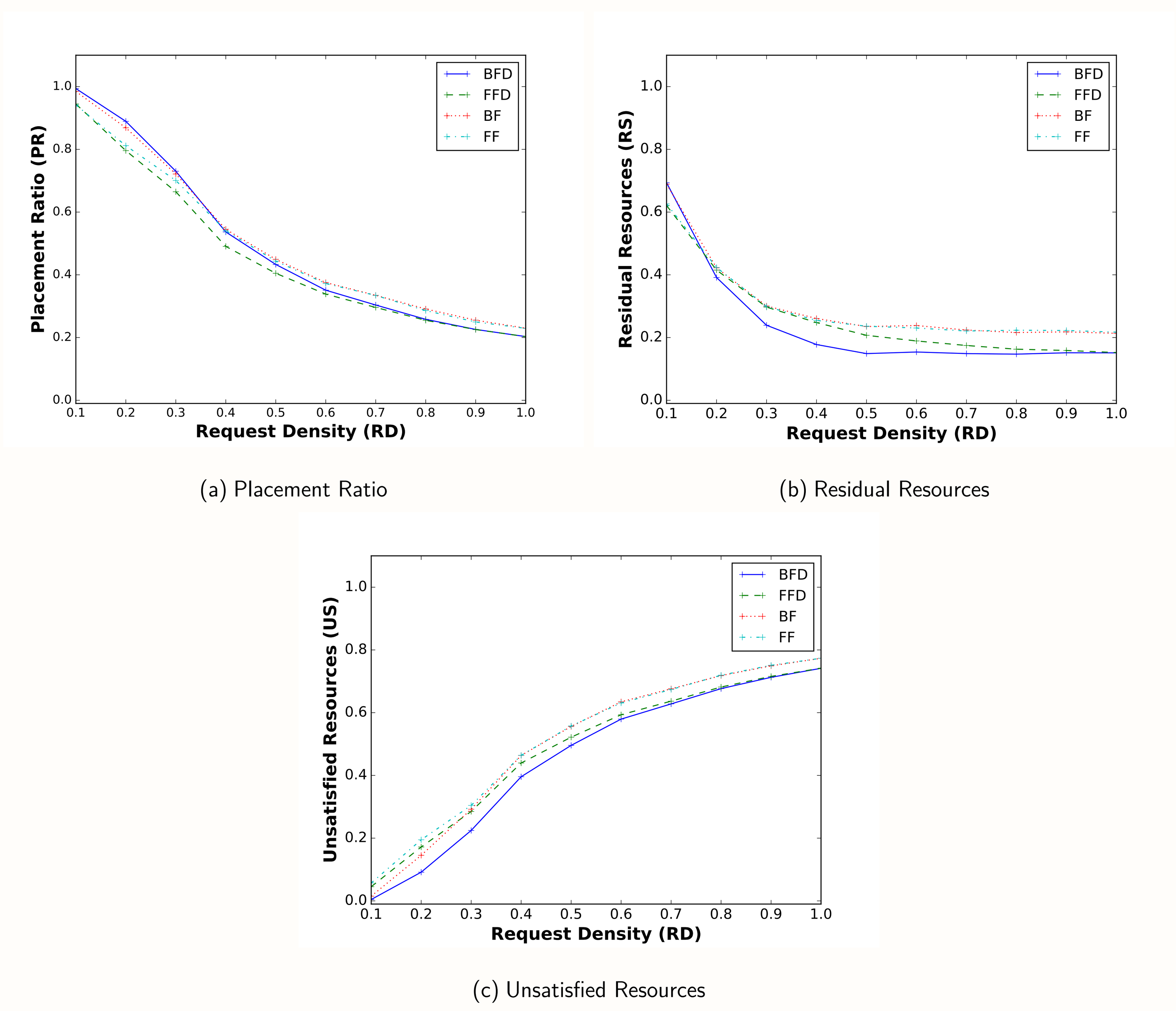

Figure 3: Evaluation of BFD, FFD, BF and FF heuristic algorithms

Conclusion

We define maximising the residual resources as an objective function for the placement problem, and propose a modified version of the BFD heuristic algorithm to be solved. We compared our version of BFD with FFD, BF and FF heuristic algorithms and demonstrate that BFD significantly reduces the unsatisfied resources compared to the other heuristic algorithms while satisfying the constraints of the problem.

Future Work

Extend the placement problem to include other security equivalence classes and enforce function chaining.

References

[1] Osanaiye, O., Choo, K.K.R. and Dlodlo, M., 2016. Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. Journal of Network and Computer Applications, 67, pp.147-165.

[2] Somani, G., Gaur, M.S. and Sanghi, D., 2015, September. DDoS/EDoS attack in cloud: affecting everyone out there!. In Proceedings of the 8th International Conference on Security of Information and Networks (pp. 169-176). ACM.

[3] Beloglazov, A., Abawajy, J. and Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, 28(5), pp.755-768.