# In-band Network Telemetry Task Orchestration based on Multi-objective Optimization

Zhenyi Zhang, Wei Su, Lizhuang Tan
*School of Electronics and Information Engineering, Beijing Jiaotong University.*
Beijing, PR China
19120173, wsu, lzhtan@bjtu.edu.cn

***Abstract*—In-band network telemetry task orchestration is to study how to reasonably select business flows to carry in-band network telemetry tasks to cover all necessary switches and ports. An inappropriate orchestration scheme not only fails to meet the requirements, but may reduce the performance of network telemetry. This paper proposes a multi-objective optimization-based in-band network telemetry task orchestration algorithm, while taking into account both the aspects of telemetry: freshness and intrusion. The experimental results show that the proposed scheme outperforms in terms of freshness and intrusion.**

***Index Terms*—In-band Network Telemetry, Network Monitoring, Multi-objective Optimization**

## I. Introduction

In-band network telemetry (INT) [1] is an advanced measurement method that has emerged in recent years. It is used in many scenarios, such as smart collaborative networks [2], [3]. Using INT to monitor the status of the entire network requires splitting and integrating the network status, monitoring the service layer, and then assigning specific telemetry items to different business flows. Marques [4] calls this process as In-band Network Telemetry Orchestration (INTO). The INTO problem involves many factors, including network topology, business flow status, and switching device port information.

Bhamare [5] proposed a telemetry framework IntOpt for monitoring the performance of Service Function Chain, and a heuristic random greedy metaheuristic based on simulated annealing algorithm. Pan [6] proposed an in-band network telemetry optimal path planning framework INT-path. The goal of INT-path planning is to minimize telemetry overhead. There are two generation algorithms for non-overlapping INT paths, depth search priority and Euler tracking. Hohemberger [7] used machine learning to solve the in-band network telemetry orchestration problem model (INTOPP). INTOPP uses the MILP model to formalize the in-band network telemetry orchestration problem, and uses the machine learning mechanism to dynamically guide the telemetry data collection process. Marques [4] proposed two single-objective optimization models. The optimization goals are to minimize the number of telemetry activity flows and minimize the saturation probability of any telemetry link. Moreover, they also proposed two corresponding heuristic solving algorithms.

However, these methods did not consider the two opposite evaluation indicators of telemetry freshness and intrusion when pursuing telemetry performance. This paper presents an in-band network telemetry task orchestration method based on multi-objective optimization, and carries out comparative experiments on NSFNET topology. The results show that it has good performance in both telemetry freshness and intrusion.

## II. System Model

In this section, we analyze the factors that affect the INTO problem, and propose two optimization objectives, and finally establish the INTO optimization model.

### A. Analysis

In-band network telemetry is carried by business packets, so the telemetry system needs to consider the constraints of potential influencing factors such as network traffic and node processing capabilities. Constraints include data link layer MTU limitation, network link remaining bandwidth limitation, switch/router service capacity limitation, and telemetry information freshness limitation, etc. When we select the telemetry flow, we need to obtain the topology of the entire network and the distribution information of each flow, as well as the telemetry demand size of each port. Moreover, the telemetry flow we finally select must cover all necessary switches and ports, and the sum of the existing data and the telemetry data of a single flow cannot exceed the MTU limit.

### B. Optimization Target

The in-band network telemetry orchestration scheme given in this paper can take into account the indicators of intrusion and freshness at the same time, which will be analyzed in turn.

*1) Intrusion:* Intrusion [8] refers to the overhead of the switch processing additional data. The switch needs to process additional data of INT, which itself will cause intrusion to the entire network. According to the principle of INT, we need to insert INT Header in each telemetry flow, and because the telemetry items required by the entire network are fixed, the intrusiveness of the INTO problem can be measured by the number of telemetry flows required to complete the telemetry task. If more flows are used, then more INT Headers need to be forwarded in the network.

Fig. 1 shows a simple case. We assume that in a network topology, there are a total of 12 ports and 4 business flows. If we need the selected telemetry flow to cover the ports of the entire network switch, we can choose f1, f2, f3, f4. However, we can also meet the requirements of covering the

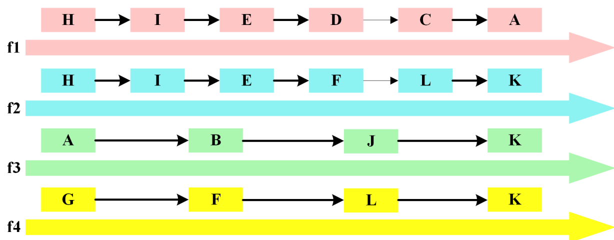


Fig. 1. Optimization example graph.

entire network switch ports without selecting all the business flows. For example, in the current topology, choosing f1, f3, and f4 can also meet the requirements, and this orchestration scheme has fewer intrusions.

*2) Freshness:* Freshness [9] refers to the speed and timeliness of the latest telemetry data obtained by upper-level telemetry applications. When we are pursuing the least amount of intrusion, it is very likely that too many telemetry projects will be concentrated in one flow, which will not only increase the forwarding processing pressure of the switch, but also increase the queue depth of the switch. At its mildest, it will increase the delay, at its worst, it will cause packet loss and throughput degradation that will reduce the freshness of the telemetry data obtained by our upper-level applications. Therefore, we need to balance the telemetry requirements of each telemetry flow to obtain better freshness.

Taking Fig. 1 as an example, assuming that the telemetry requirements of each port are the same, then ideally we hope that each telemetry flow is allocated 12/4=3 ports, so we can choose such an orchestration scheme: assign f1, f2, f3, and f4 {C, D, E}, {H, I, K}, {A, B, J} and {F, G, L} respectively. But the actual problem is much more complicated than this, because the number of telemetry requirements for each port is different, and the size of each telemetry requirement is also different.

### C. Optimization Model

In this subsection, we give an optimization model for in-band network telemetry orchestration related to network topology information and network traffic information.

The INTO problem can be expressed as a physical network $G$=($D$,$P$) and a set of business flows $F$. The set $D$ represents the programmable forwarding device in the network $D$={1,2,...,$|D|$}. Each device $d{\in}D$ has a set of network ports to connect to other network devices. The set $P$ represents the set of device ports in the network, and there is a telemetry requirement for each port $p{\in}P$. The number of telemetry item bits that flow $F$ needs to collect regularly is $\delta(p){\in}N^+$

The optimization model can be described as $\Phi$:$P{\rightarrow}F$, where $P$ is the device port set, $F$ is the business flow set, the variable $x_{p,f}$ represents whether the flow $f$ should cover the port $p{\in}P$, and the value of the variable $x_{p,f}$ represents the orchestration plan $\Phi$. $x_{p,f} = 1$ means that the device port $p$ is assigned to the flow $f$. The variable $y_f = 1$ means that the flow $f$ is selected as the telemetry flow. Each flow $f{\in}F$ has two endpoints, which are transmitted in a fixed path in the network $G$. We define the available telemetry capacity of each flow $f$ as $\varepsilon(f){\in}N^+$, which represents the sum of the maximum telemetry item bits that each packet can carry.

Therefore, we give the following optimization model representation:

$$min\Phi_1 = \sum_{f\in F} y_f \tag{1}$$

$$min\Phi_2 = k \tag{2}$$

$$s.t. \sum_{f\in F} x_{p,f} = 1, \forall p \in P, \delta(p) > 0 \tag{3}$$

$$\sum_{i\in\rho(f)} x_{p,f} \cdot \delta(i) \leq y_f \cdot \varepsilon(f), \forall f \in F \tag{4}$$

$$x_{p,f} \in \{0,1\}, \forall p \in P, \forall f \in F \tag{5}$$

$$y_f \in \{0,1\}, \forall f \in F \tag{6}$$

$$k > 0 \tag{7}$$

The objective function (1) represents the minimization of the number of telemetry flows. The variable $k$ in the objective function (2) represents the sum of the maximum telemetry item bits allocated to the telemetry item carried by a single flow, which means that the maximum telemetry load of each telemetry flow is the smallest. Use this function to represent all flows to obtain as balanced telemetry demand as possible. Constraint condition (3) guarantees that all ports $p{\in}P$ are covered by a certain flow $f{\in}F$. Constraint (4) means that the number of telemetry items that need to be collected on each telemetry flow does not exceed the limit of its capacity $\varepsilon(f)$. Constraints (6) and (7) define the scope of $x_{p,f}$, $y_f$ and $k$.

## III. Scheme Design

The algorithm designed in this scheme is based on the non-dominated sorting genetic algorithm(NSGA-II), in which the mutation operation is improved according to the characteristics of the INTO problem, and the pre-repair population operation is performed after each cross and mutation operation. The first input of this algorithm is the information of all business flows existing in the network topology. We uniquely number all ports in the network, and represent one business flow information through the port numbers flowing through; The second is the remaining available capacity of each business flow; The third is the telemetry requirements of each port. The process of this algorithm is shown in Fig. 2. This section will explain the important parts of the method.

### A. Chromosome Coding

The chromosome length N is the sum of the number of ports passed by each flow, and the coding method adopts the 0-1 backpack idea, that is, the numbers at the N positions indicate whether the port corresponding to the position has a business flow to carry its telemetry needs. For example, there are two business flows, and the port numbers they pass through are [1,2,5,8] and [10,2,7,3]. The first flow carries the

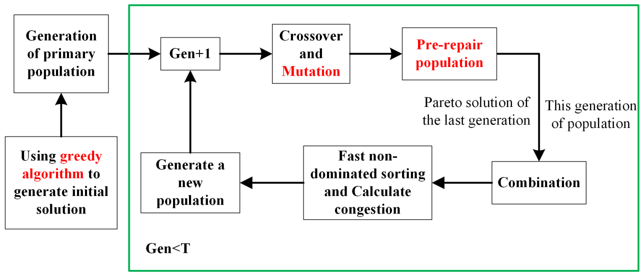


Fig. 2. The process of algorithm.

telemetry demand of ports 1, 2, and 8, the second flow carries the telemetry demand of ports 7 and 3. Then the chromosome code of an individual is [1,1,0,1,0,0,1,1]. The 1 in the first place indicates that the first flow carries the telemetry demand of the first port through which it flows, that is, the telemetry demand of port 1, and the meaning of other positions can be deduced by analogy.

### B. Initial Solution Construction

We used the greedy algorithm for the construction of the initial solution, because all necessary ports need to be covered at the same time and only once, we actually need to select some ports from each flow to meet the above requirements. The scheme of the greedy algorithm to select ports is as follows:

(1) Sort according to the demand size of the port from large to small.

(2) Traverse the sorted ports in turn to find the business flow through the port.

(3) Select the business flow with the largest remaining capacity among these business flows.

(4) Update the loading capacity and remaining capacity of the selected business flow.

The constructed initial solution can satisfy the constraint that all necessary ports need to be covered and covered only once.

### C. Crossover and Mutation

For the crossover operation, it is relatively simple. The method uses two-point crossover to generate two random integers, and then intercept the chromosome in the middle of these two positions for exchange. After the crossover operation, we perform the pre-repair population operation because there will be individuals who do not meet the basic constraints after each crossover operation, that is, the full port coverage is not met or some ports are covered multiple times. The repair operation makes it meet the two conditions above, which can improve the efficiency of evolution.

For the mutation operation, we want the population to mutate towards the goal of freshness. For an individual mutation operation, we take out any port on a business flow with the most telemetry items and let another business flow carry it. After the mutation operation, the pre-repair population operation is also carried out.

### D. Algorithm Process

To sum up, the operation process of the algorithm used in this scheme is shown in Fig. 2:

(1) Use the greedy algorithm to generate an initial solution and an initial population $P_t$ with a population size of $N$ according to the initial solution.

(2) $P_t$ forms an offspring population $Q_t$ of size $N$ through improved crossover and mutation.

(3) Pre-repair the population after crossover and mutation.

(4) Include this generation of population and the pareto solution of size $M$ generated by the previous generation into the evolution pool. Now there are a total of $M+N$ individuals.

(5) Reclassify the $M+N$ individuals by non-inferior grade according to the calculated objective function values.

(6) Calculate the local crowded distance of individuals at each level and select $N$ individuals as the new population $P_t$.

(7) Repeat steps (2)$\sim$(6) when the number of iterations is less than $T$.

## IV. Experimental Results

We build NSFNET topology to test our results and compare with two single objective algorithms.

### A. Environment

The topology used in this experiment is NSFNET network topology, which contains 14 network nodes, 21 links, and 42 ports. We numbered the ports as shown in Fig. 3:

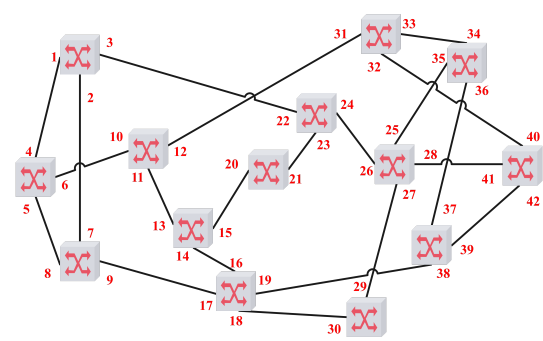

Fig. 3. Topology of NSFNET.

According to the cost of each link, we generate 91 business flows by Dijkstra algorithm. Telemetry items include queue timestamp (48bits), packet queuing delay (32bits), switch ID (8bits), queue depth (16bits) and other information. Because each telemetry item needs a different capacity, we choose the average value of 26bits as the telemetry required capacity of each telemetry item. For each port, telemetry requirement items are randomly selected from 4 to 10 items, following a uniform distribution. This range is selected because four items can be considered as the minimum items needed to measure a piece of port information (switch ID + port ID + queue timestamp + queue delay / queue depth), while 10 is the number of common metadata fields that can be exported by the device according to [10]. For the remaining available telemetry capacity of each flow, we randomly select it according to

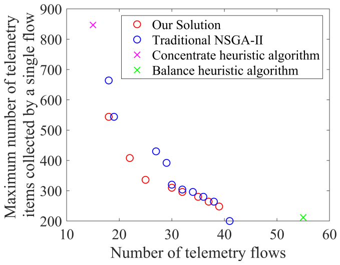


Fig. 4. Comparison of the results of four schemes.

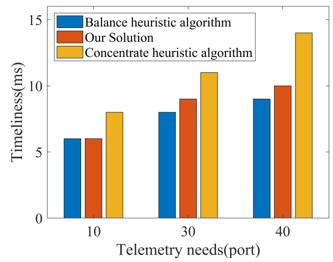


Fig. 5. Average telemetry time of three schemes.

the normal distribution mean equals 35 terms and standard deviation equals 5 terms.

### B. Result Analysis

The parameters in our experiment are: the population size is 100, the number of iterations is 300, the crossover probability is 10%, and the mutation probability is 5%. At the end of the iteration, a set of Pareto frontier solutions is generated, and each solution in the solution set corresponds to an in-band network telemetry scheduling scheme applied to the current topology. The Pareto frontier solution set contains 8 solutions. The Pareto front graph is shown as red circles in Fig. 4.

Compared with the random mutation and the direct elimination of the solutions that do not meet the basic constraints after the cross mutation, the improved mutation operation and the scheme of repairing the population in advance can selectively achieve the freshness goal, and find a better solution within a limited number of iterations. As shown in Fig. 4, the results of running with the same parameters, blue represents the results of the traditional NSGA-II algorithm, and red represents the results of our improved NSGA-II algorithm. It can be seen that the improved NSGA-II algorithm can find better solutions than the traditional algorithm.

We compare this scheme with the centralized optimization scheme with the single optimization objective of minimizing telemetry flow [4] and the balance optimization scheme with the single optimization objective of average telemetry flow carrying project. The results are shown in Fig. 4. It can be seen that for the centralized optimization of a single target, although the number of business flows used in the final telemetry orchestration solution is less than that obtained by our algorithm, the corresponding telemetry requirements that each telemetry flow needs to carry are much larger. Then the forwarding pressure of the corresponding telemetry flow through the switch will be increased, and the freshness of the telemetry flow measurement data will decrease. For single-objective balance optimization, the number of business flows used by the orchestration solution will increase significantly, which will bring great intrusion to the network and affect the forwarding efficiency. These two goals are mutually restrictive. Our solution can effectively balance these two goals and give a set of feasible and excellent solutions that users can choose according to their preferences.

We compare the average telemetry completion time of the three schemes under different telemetry requirements. For simplicity, we think that each port corresponds to a telemetry requirement. The result is shown in Fig. 5. The average telemetry time of balanced optimization is faster because it does not consider the problem of invasion, but our scheme is not far behind. It can be seen that with the increase of telemetry demand in the whole network, the telemetry completion time of the centralized optimization scheme is much longer than our scheme.

## V. Conclusion

This paper analyzes the problem of in-band network telemetry orchestration, proposes two optimization indicators based on telemetry intrusion and telemetry freshness, and proposes an in-band network telemetry orchestration algorithm based on multi-objective optimization, which can take into account both the above two indicators at the same time.

## Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant No.2018YFB1800305 and the National Natural Science Foundation of China under Grant 62072030 and the Fundamental Research Funds for the Central Universities of China under Grant No. 2020YJS013. (Corresponding Author: Wei Su).

## References


[1] L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, "In-band network telemetry: A survey," *Computer Networks*, vol. 186, p. 107763, 2021.

[2] F. Song, M. Zhu, Y. Zhou, I. You, and H. Zhang, "Smart collaborative tracking for ubiquitous power iot in edge-cloud interplay domain," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6046–6055, 2019.

[3] F. Song, Z. Ai, Y. Zhou, I. You, K.-K. R. Choo, and H. Zhang, "Smart collaborative automation for receive buffer control in multipath industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1385–1394, 2019.

[4] J. A. Marques, M. C. Luizelli, R. I. T. da Costa Filho, and L. P. Gaspary, "An optimization-based approach for efficient network monitoring using in-band network telemetry," *Journal of Internet Services and Applications*, vol. 10, no. 1, pp. 1–20, 2019.

[5] D. Bhamare, A. Kassler, J. Vestin, M. A. Khoshkholghi, and J. Taheri, "Intopt: In-band network telemetry optimization for nfv service chain monitoring," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[6] T. Pan, E. Song, Z. Bian, X. Lin, X. Peng, J. Zhang, T. Huang, B. Liu, and Y. Liu, "Int-path: Towards optimal path planning for in-band network-wide telemetry," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 487–495.

[7] R. Hohemberger, A. G. Castro, F. G. Vogt, R. B. Mansilha, A. F. Lorenzon, F. D. Rossi, and M. C. Luizelli, "Orchestrating in-band data plane telemetry with machine learning," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2247–2251, 2019.

[8] F. Song, Z. Ai, H. Zhang, I. You, and S. Li, "Smart collaborative balancing for dependable network components in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, 2020.

[9] N. S. Kagami, R. I. T. da Costa Filho, and L. P. Gaspary, "Capest: Offloading network capacity and available bandwidth estimation to programmable data planes," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 175–189, 2019.

[10] The P4.org Applications Working Group, "In-band network telemetry(int) dataplane specification version 2.0," https://github.com/p4lang/p4-applications/blob/master/docs/INT_v2_0.pdf/.