Compressing two-dimensional routing tables with order

Frédéric Giroire, Frédéric Havet, Joanna Moulierac

Compressing Two-dimensional Routing Tables with Order ⋆

Frédéric Giroire [a,1] Frédéric Havet [a] Joanna Moulierac [a]

[a] *CNRS, Laboratoire I3S, UMR 7172, UNS, Inria, Coati Sophia Antipolis, France*

Abstract

A *communication* in a network is a pair of nodes (s, t). The node s is called the source *source* and t the *destination*. A *routing* of a communication (s, t) is a path in the network from s to t. A *routing* of a communication set is a union of routings of its communications.

At each node, there is a set $\mathcal{X}$ of communications whose routing path goes through this node. The node needs to find for each communication (s, t) in $\mathcal{X}$, the port that the routing path of (s, t) uses to leave it. An easy way of doing it is to store the list of all triples (s, t, k), where $(s, t) \in \mathcal{X}$ and k is the port used by the (s, t)-path to leave the node.

However, such a list might be very large. Motivated by routing in telecommunication network using Software Defined Network (SDN) technologies, we consider the problem of compacting this list using aggregation rules. Hence, in addition, we can use some additional triples, called $*$*-triples*. As an example, a t*-destination triple* $(*, t, p)$, means that every communication with destination t leaves on port p.

We carry out in this work a study of the problem complexity, providing results of NP-completeness, of Fixed-Parameter Tractability and approximation algorithms.

Keywords: routing, compact routing tables, software defined networks, complexity, approximation algorithms

⋆ This work has been partially supported by ANR project Stint under reference ANR-13-BS02- 0007, ANR program Investments for the Future under reference ANR-11-LABX-0031-01, ANR VISE, CNRS-FUNCAP project GAIATO, the associated Inria team AlDyNet, the project ECOS-Sud Chile.

[1] Corresponding author. Email: `frederic.giroire@cnrs.fr`

1 Introduction

Motivation. Software Defined Technology (SDN), e.g. OpenFlow [13] is a new promising approach to operate telecommunication networks. Its promise is to allow dynamic routing decisions by decoupling the control plane (the system making decisions) from the data plane (which forwards the packets). This way, a centralized controller receives the data monitored in the system (e.g. load, delay, ...) and then, based on this information, computes appropriate routing decisions, e.g. to improve energy efficiency [5,7]. Each time a new flow arrives, the router contacts the controller and waits for the decisions to be pushed into its forwarding table. The routing tables thus are populated with flow-based rules with header informations (source IP, destination IP, ...) $\rightarrow$ exit port.

However, SDN hardware uses specific memory, e.g. TCAM memory [9,10], which is very expensive and of small size. Thus, the number of entries of the routing tables is limited to only a few thousands [14,15] and grows linearly with the number of flows passing through a router, causing a problem of scalability. It is thus an important area of research to obtain routing using only a limited number of rules per router. [2] studies the problem of choosing routing with a limited number of entry per router using linear programming. Another way to compact the forwarding tables is to use aggregation rules. With such an aggregation, we can set routing entries such as "(*,destination)$\rightarrow$ port" or "(source,*)$\rightarrow$ port" or also a default entry such as "(*,*)$\rightarrow$ port". For example, [6] studies how to use default ports to reduce the size of routing tables. In this work, we consider the problem of compressing a routing table using aggregation rules.

We consider here *two-dimensional routing tables* in which the routing decision is not done exclusively on the destination IP address, but on the source and destination IP addresses. Indeed, the commonly implemented destination-based routing has its limitations, especially in delivering quality of service which is a goal of SDN paradigm. One suggested remedy is to base the routing decision on additional fields in the packet header. One of the most important field is the source host. For instance, this would permit selective routing to provide a high bandwidth connection between two different sites of a company. Such refined forwarding is part of the next generation Internet design, and falls within the broader scope of layer four packet classification, where packets are routed using arbitrary fields of the packet header [8], [3], [12], [1]. Routers capable of packet classification can implement many advanced services, such as firewall access control, Virtual Private Networks, and quality

of service routing, which are all promises of the SDN paradigm.

Modeling. A *communication* in a network is a pair of nodes (s, t). The node s is called the source *source* and t the *destination*. We use the source and destination fields in our examples, although our ideas apply to any two prefix fields in Internet protocol networks. A *communication set* is a set of distinct communications, i.e. two communications might have the same source or the same destination, but they cannot have both same source and same destination. A *routing* of a communication (s, t) is a path in the network from s to t. A *routing* of a communication set is a union of routings of its communications.

At each node, there is a set $\mathcal{X}$ of communications whose routing path goes through this node. The node needs to find for each communication (s, t) in $\mathcal{X}$, the port that the routing path of (s, t) uses to leave it. An easy way of doing it is to store the list of all triples (s, t, k), where $(s, t) \in \mathcal{X}$ and k is the port used by the (s, t)-path to leave the node. Such triples are called *communication triples*.

However, such a list might be very large. So we want to reduce it as much as possible using the $*$ symbol. Hence, in addition, we can use some additional triples, called $*$*-triples*. There are two kinds of $*$-triples:

- t*-destination triple* $(*, t, p)$, meaning that every communication with destination t leaves on port p.
- s*-source triple* $(s, *, p)$, meaning that every communication with source t leaves on port p.

A *routing list* is an ordered list $T_1, \ldots, T_r$ of triples (either communication, or source, or destination ones). A communication is then assigned the port of the *first* triple in the list, that applies to it. It is crucial to remark that *using $*$-triples introduces an order of the rules in the routing list.*

Let $\mathcal{C}$ be a set of communication triples. A routing list $\mathcal{R}$ *emulates* $\mathcal{C}$ if each communication of $\mathcal{C}$ is assigned the same port by $\mathcal{C}$ and $\mathcal{R}$. Observe that $\mathcal{R}$ may route more communications than $\mathcal{C}$. For example, if the port of all triples of $\mathcal{C}$ have source s and port p, then the singleton list made of the source triple $(s, *, p)$ emulates $\mathcal{C}$, even if there is not a triple in $\mathcal{C}$ for all communications.

Problems. The problem is then to find the shortest routing list that emulates $\mathcal{C}$. We denote by $\mathrm{rmin}(\mathcal{C})$ the minimum number of triples in a routing list emulating $\mathcal{C}$.

Routing List:

Input: A set $\mathcal{C}$ of communication triples and an integer r.
Question: $\text{rmin}(\mathcal{C}) \leq \text{r}$?

The *number of saved triples* is $\text{sav}(\mathcal{C}) = |\mathcal{C}| - \text{rmin}(\mathcal{C})$. The complementary problem to Routing List is the following.

List Reduction
Input: A set $\mathcal{C}$ of communication triples and an integer z.
Question: $\text{sav}(\mathcal{C}) \geq \text{z}$?

Contributions. In this work, we study the complexity of the above problems. We provide NP-completeness results and an approximation algorithm. Due to lack of space, some proofs are omitted. Full proofs can be found in [4].

Our work answers an open question of [16]. Similarly to us, the authors consider the problem of determining a compact routing table using aggregation rules that has the same behavior as the original routing table. The difference with our problem is that their goal is to find what they call a *conflict-free* routing table in which the rules can be taken in any order. On the contrary, as noted above, the order is crucial in our problems.

2 Results

We show that Routing List and List Reduction are NP-complete by reduction from Feedback Arc Set problem, one of Karp's 21 NP-complete problems [11].

Theorem 2.1 Routing List *and* List Reduction *are NP-complete, even if there are only two ports.*

Because of the NP-hardness of the problem it is interesting to get approximation algorithms. We show that sav can be 2-approximated in polynomial time using the simple heuristic described below:

The Direction-based Heuristic Let $\mathcal{C}$ be a set of communication triples with destination set S and destination set T. Set $n = |S|$ and $m = |T|$. For any port p, let $\mathcal{C}(p)$ be the set of triples of $\mathcal{C}$ with port p. For a source s (resp. destination t) and a port p, let $\mathcal{C}(s,p)$ (resp. $\mathcal{C}(t,p)$) be the set of triples of $\mathcal{C}$ with source s (resp. destination t) and port p. For every source s, let $M(s) := \max_p |\mathcal{C}(s,p)|$ be the maximum number of triples in $\mathcal{C}$ with source s and same port, and for every destination t, let $M(t) := \max_p |\mathcal{C}(t,p)|$ be the maximum number of triples in $\mathcal{C}$ with destination t and same port. Set

$$Z^-(\mathcal{C}) = \sum_{s\in S}(M(s)-1) = \sum_{s\in S} M(s) - n \quad \text{and}$$
$$Z^|(\mathcal{C}) = \sum_{t\in T}(M(t)-1) = \sum_{t\in T} M(t) - m.$$

Any routing list emulating $\mathcal{C}$ yields an upper bound on $\mathrm{rmin}(\mathcal{C})$. One such list can be obtained by routing source by source. One source s after another we route all triples of $\mathcal{C}$ with source s. This can be done by using the triple $(s, *, p)$ for p a port such that there are $M(s)$ triples with source s and port p after all triples with source s and port distinct from p. Doing so, we save $M(s)-1$ triples when routing the triples with source s. Hence, we obtain a routing list of size $|\mathcal{C}| - Z^-(\mathcal{C})$. Such a list is called a *source-based routing list*.

Proceeding similarly according to the destinations, we obtain a routing list, called *destination-based* of size $|\mathcal{C}| - Z^|(\mathcal{C})$.

Setting $Z(\mathcal{C}) = \max\{Z^-(\mathcal{C}), Z^|(\mathcal{C})\}$, we have

$$\mathrm{sav}(\mathcal{C}) \geq \mathrm{Z}(\mathcal{C}) \text{ and } \mathrm{rmin}(\mathcal{C}) \leq |\mathcal{C}| - \mathrm{Z}(\mathcal{C}). \tag{1}$$

The algorithm consisting in computing a source-based routing list and a destination-based routing list and taking the shortest of the two, is called the *Direction-based Heuristic*. It provides a routing list emulating $\mathcal{C}$ of size $Z(\mathcal{C})$. The Direction-based Heuristic is a 2-approximation for List Reduction.

Theorem 2.2 *The Destination-based Heuristic is a* 2*-approximation for* List Reduction.

Proof. Let $\mathcal{C}$ be a set of communication triples with destination set S and destination set T. Set $n = |S|$ and $m = |T|$. We can order S and T by decreasing order according to the function M. That is $M(s_1) \geq M(s_2) \geq \cdots \geq M(s_n)$ and $M(t_1) \geq M(t_2) \geq \cdots \geq M(t_m)$.

The directed $\{0, 1, \ldots, n\} \times \{0, 1, \ldots, m\}$-grid, denoted by $G_{n,m}$ is the digraph defined by

$$V(G_{n,m}) = \{(i,j) \mid 0 \leq i \leq n \text{ and } 0 \leq j \leq m\}$$
$$A(G_{n,m}) = A^h(G_{n,m}) \cup A^v(G_{n,m}),$$

where $A^h(G_{n,m}) = \{a^h(i,j) \mid 1 \leq i \leq n \text{ and } 0 \leq j \leq m\}$, with $a^h(i,j) = ((i-1,j),(i,j))$, is the set of *horizontal arcs*, and $A^v(G_{n,m}) = \{a^v(i,j) \mid 0 \leq i \leq n \text{ and } 1 \leq j \leq m\}$, with $a^v(i,j) = ((i,j-1),(i,j))$, is the set of *vertical arcs*.

An *edge-weighted digraph* is a pair (G, w) where G is a digraph and w a *weight function* on its arcs, that is a function from $A(D)$ into $\mathbb{R}$. The *length*

of a path U in an edge-weighted digraph (G, w) is the sum of the weights of its arcs: $w(U) = \sum_{a \in A(U)} w(a)$.

Let $w_{\mathcal{C}}$ be the weight function defined on $A(G_{n,m})$ by $w_{\mathcal{C}}(a^h(i,j)) = \min\{M(s_i), m-j\}-1$ and $w_{\mathcal{C}}(a^v(i,j)) = \min\{M(t_j), n-i\}-1$. Let $W(\mathcal{C})$ be the maximum length of a path in $(G_{n,m}, w_{\mathcal{C}})$. Observe that it is attained by a path from $(0,0)$ to one of the sides $\{n\} \times \{0,1,\ldots,n\}$ and $\{0,1,\ldots,n\} \times \{n\}$ because the weight of an arc is negative if and only if it is in $\{a^h(i,m) \mid 1 \leq i \leq n\} \cup \{a^v(n,j) \mid 1 \leq j \leq m\}$.

We shall prove that

$$\mathrm{sav}(\mathcal{C}) \leq \mathrm{W}(\mathcal{C}). \tag{2}$$

Let $\mathcal{R}$ be a routing list that emulates $\mathcal{C}$. One can show (see [4]) that we can freely rearrange the triples of $\mathcal{R}$, so that $\mathcal{R}$ is canonical. A routing list is *canonical* if it is the concatenation of sublist $\mathcal{B}_1, \ldots, \mathcal{B}_q$, called *blocks* having the following properties for every $1 \leq \ell \leq q$:

(i) in $\mathcal{B}_\ell$, there is a unique $*$-triple and it is the last one;

(ii) if the $*$-triple of $\mathcal{B}_\ell$ is an s-source triple (resp. t-destination triple), then all triples of $\mathcal{B}_\ell$ have source s (resp. destination t).

For $1 \leq k \leq q$, let T_{ℓ_k} be the $*$-triple of $\mathcal{B}_k$, let r_k be the number of triples of $\mathcal{C}$ routed by T_{ℓ_k}, and let i_k (resp. j_k) be the number of source triples (resp. destination triples) with index at most ℓ_k. We have $|\mathcal{R}| = |\mathcal{C}| - \Sigma$ with $\Sigma = \sum_{k=1}^{q} (r_k - 1)$.

Let K_s (resp. K_t) be the set of indices k such that T_{ℓ_k} is a source (resp. destination) triple. For $k \in K_s$, s'_k be the source of T_{ℓ_k}, and for $k \in K_t$, t'_k be the destination of T_{ℓ_k}.

Assume $k \in K_s$. When routing according to $\mathcal{R}$, before considering the block $\mathcal{B}_k$, there are at most $m - j_k$ triples of $\mathcal{C}$ with source s'_k and at most $M(s'_k)$ triples of $\mathcal{C}$ with source s'_k. Therefore, $r_k \leq \min\{M(s'_k), m - j_k\}$. Similarly, if $k \in K_t$, then $r_k \leq \min\{M(t'_k), n - i_k\}$. Thus

$$\Sigma \leq \sum_{k \in K_s} (\min\{M(s'_k), m - j_k\} - 1) + \sum_{k \in K_t} (\min\{M(t'_k), n - i_k\} - 1).$$

Set $u_0 = (0,0)$ and $u_k = (i_k, j_k)$ for $1 \leq k \leq q$. Note that $i_k + j_k = k$ for all $0 \leq k \leq q$. The sequence $U = u_0, u_1, \ldots, u_q$ can be seen as a path in the directed grid $G_{n,m}$. Observe that if $k \in K_s$ then $u_{k-1}u_k$ is a horizontal arc and $w_{\mathcal{C}}(u_{k-1}u_k) = \min\{M(s_{i_k}), m - j_k\} - 1$, and if $k \in K_t$ then $u_{k-1}u_k$ is a

vertical arc and $w_{\mathcal{C}}(u_{k-1}u_k) = \min\{M(t_{j_k}), n - i_k\} - 1$. Hence

$$w_{\mathcal{C}}(U) = \sum_{k \in K_s} (\min\{M(s_{i_k}), m - j_k\} - 1) + \sum_{k \in K_t} (\min\{M(t_{j_k}), n - i_k\} - 1).$$

Now since $M(s_1) \geq M(s_2) \geq \cdots \geq M(s_n)$ and $j_1 \leq j_2 \leq \cdots \leq j_q$, we have $\sum_{k \in K_s} (\min\{M(s'_k), m - j_k\} - 1) \leq \sum_{k \in K_s} (\min\{M(s_{i_k}), m - j_k\} - 1)$. Similarly, $\sum_{k \in K_t} (\min\{M(t'_k), n - i_k\} - 1) \leq \sum_{k \in K_t} (\min\{M(t_{j_k}), n - i_k\} - 1)$. Hence $\Sigma \leq w_{\mathcal{C}}(U)$. Thus $\Sigma \leq W(\mathcal{C})$, which implies (2).

One can check (see [4]) that $W(\mathcal{C}) \leq 2Z(\mathcal{C})$, yielding the theorem. □

3 Conclusion

We studied the complexity of Routing List and of List Reduction. We provide results of NP-completeness and an approximation algorithm.

We leave several questions as open problems:

- The Destination-based Heuristic is a 2-approximation for List Reduction. However, the heuristic often returns a routing list that saves more than a half of $\text{sav}(\mathcal{C})$ triples. This leads to think that the approximation of 2 is not best possible.

  Problem 3.1 *What is the best approximation ratio for* List Reduction*?*

- We proved the NP-Completeness of Routing List by a reduction to Feedback Arc Set. Knowing that Feedback Arc Set is APX-complete,

  Problem 3.2 *Is* Routing List *also APX-complete? For which approximation ratio?*

References

[1] Buddhikot, M. M., S. Suri and M. Waldvogel, "Space decomposition techniques for fast layer-4 switching," Springer, 2000.

[2] Cohen, R., L. Lewin-Eytan, J. Naor and D. Raz, *On the effect of forwarding table size on SDN network utilization*, in: *IEEE INFOCOM*, 2014, pp. 1734–1742.

[3] Eppstein, D. and S. Muthukrishnan, *Internet packet filter management and rectangle geometry*, in: *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, 2001, pp. 827–835.

[4] Giroire, F., F. Havet and J. Moulierac, *Compressing Two-dimensional Routing Tables with Order*, Research Report RR-8658, INRIA Sophia Antipolis (2014).

[5] Giroire, F., D. Mazauric, J. Moulierac and B. Onfroy, *Minimizing routing energy consumption: from theoretical to practical results*, in: *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on*, IEEE, 2010, pp. 252–259.

[6] Giroire, F., J. Moulierac and T. Khoa Phan, *Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing*, in: *IEEE GLOBECOM*, IEEE, Austin Texas, United States, 2014.

[7] Giroire, F., J. Moulierac, T. K. Phan and F. Roudaut, *Minimization of network power consumption with redundancy elimination*, in: *NETWORKING 2012*, Springer, 2012 pp. 247–258.

[8] Hari, A., S. Suri and G. Parulkar, *Detecting and resolving packet filter conflicts*, in: *INFOCOM 2000*, IEEE, 2000, pp. 1203–1212.

[9] Kang, N., Z. Liu, J. Rexford and D. Walker, *Optimizing the "one big switch abstraction" in software-defined networks*, in: *Proceedings of CoNEXT* (2013), pp. 13–24.

[10] Kanizo, Y., D. Hay and I. Keslassy, *Palette: Distributing tables in software-defined networks*, in: *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 545–549.

[11] Karp, R. M., "Reducibility among combinatorial problems," Springer, 1972.

[12] Lakshman, T. and D. Stiliadis, *High-speed policy-based packet forwarding using efficient multi-dimensional range matching*, ACM SIGCOMM Computer Communication Review **28** (1998), pp. 203–214.

[13] McKeown, N., T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, *Openflow: Enabling innovation in campus networks*, SIGCOMM Comput. Commun. Rev. **38** (2008), pp. 69–74.

[14] Narayanan, R., S. Kotha, G. Lin, A. Khan, S. Rizvi, W. Javed, H. Khan and S. Khayam, *Macroflows and microflows: Enabling rapid network innovation through a split sdn data plane*, in: *Software Defined Networking (EWSDN), 2012 European Workshop on*, 2012, pp. 79–84.

[15] Stephens, B., A. Cox, W. Felter, C. Dixon and J. Carter, *Past: Scalable ethernet for data centers*, in: *Proceedings of CoNEXT* (2012), pp. 49–60.

[16] Suri, S., T. Sandholm and P. Warkhede, *Compressing two-dimensional routing tables*, Algorithmica **35** (2003), pp. 287–300.