Future Generation Computer Systems 102 (2020) 1–13

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# Congestion avoidance transmission mechanism based on two-dimensional forwarding

Wenlong Chen [a], Heyang Chen [a], Zhiliang Wang [b,e,*], Chengan Zhao [c], Mingwei Xu [d,e], Ke Xu [d,e], Yingya Guo [d,e]

[a] College of Information Engineering, Capital Normal University, Beijing 100048, China
[b] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China
[c] School of Management, Capital Normal University, Beijing 100048, China
[d] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[e] Beijing National Research Center for Information Science and Technology, Beijing 100084, China

## HIGHLIGHTS

- Because of the optimized algorithm, CATD realizes congestion avoidance rapidly.
- CATD makes full use of the other links so as to enhance the bandwidth utilization of the whole network.
- The constructed CA-Path is obviously shorter than the path of the other mechanism.
- The calculation of CA-Path employs those links which will not be congested after bypassing so CATD will not lead to more congestion.

## ARTICLE INFO

*Article history:*
Received 19 December 2018
Received in revised form 19 June 2019
Accepted 25 July 2019
Available online 29 July 2019

*Keywords:*
Link congestion
Shortest path
Two-dimensional routing
CA-path
CA-flow

## ABSTRACT

In the existing intra-domain routing mechanism, IP packets are transmitted along the shortest path, which frequently leads to congestion of some key links or make some links in danger of congestion (dangerous link). Multipath transmission mechanisms are often adopted to settle such problem, but they have disadvantages such as extra packet field and lack of real-time response. In this paper, the Congestion Avoidance transmission mechanism based on Two-Dimensional routing (CATD) is proposed. When congestion is about to occur in a link, a congestion avoidance path (CA-Path) is constructed based on two-dimensional forwarding to bypass part of the traffic of this link to other paths. We also put forward the optimal scheme of CA-Path, which avoids redundant transmission and reduces the deployment of two-dimensional entries. Furthermore, in order to minimize the number of two-dimensional entries, we propose the selection rules of congestion avoidance flow (CA-Flow). The congestion avoidance based on two-dimensional routing has four advantages that: it can divide all the traffic more meticulously based on their sources and destinations, it avoids conflict with previous one-dimensional forwarding, it brings no extra field in packet, and it will not lead to subsequent link congestion because of the deployment of SDN (Software Defined Network) server. Based on actual network and global typical topology, our experiments show that CATD realizes congestion avoidance rapidly and enhances the bandwidth utilization of the whole network, and the CA-Path is shorter than that of the other mechanism.



## 1. Introduction

Traditional intra-domain network routing usually implements packet transmission along the shortest path. However, with the enlargement of network scale, network link congestion frequently occurs in the current strategy especially in key links. Multipath transmission is the main approach to congestion avoidance, since nearly 75% links in network are available to offload some traffic from the hot-spot links [1]. Whereas, there are still some drawbacks in multipath mechanism, such as the label encapsulation costs of MPLS (Multi-Protocol Label Switching) [2] and redundant storage for back-up paths.

This paper proposes Congestion Avoidance transmission mechanism based on Two-Dimensional forwarding (CATD), which transmits some of the traffic in dangerous link to other paths. In the existing network, each node counts its link bandwidth per interface and reports it to the SDN [3,4] server. In CATD, when the link load is more than one specific threshold (set

* Corresponding author.
*E-mail address:* wzl@cernet.edu.cn (Z. Wang).

https://doi.org/10.1016/j.future.2019.07.057
0167-739X/

by the administrator), this link will be recognized as dangerous and may cause congestion. Then the upstream node of the link constructs a congestion avoidance path (CA-Path) based on two-dimensional forwarding to bypass part of the traffic on the link. Furthermore, the optimal scheme of CA-Path is given to eliminate redundant forwarding and decrease the storage of two-dimensional forwarding entries. Two-dimensional routing marks the routing items with the source and destination addresses, which enables the router to provide different transmission paths for packets going to the same destination address according to different source addresses. CATD implements congestion avoidance based on two-dimensional routing, which has three main advantages: (1) it subdivides traffic in detail and controls flows in dangerous link accurately; (2) since two-dimensional entry is different from previous entry, the later deployment for congestion avoidance will not conflict with preceding forwarding; (3) the direct employment of two-dimensional entry reduces the change to existing protocol and forwarding packet. According to the mechanism above, this paper further proposes a strategy to select congestion avoidance flow, which mainly chooses the collection of flow that not only satisfies the demand of bandwidth reduction but also minimizes the calculation burden of CA-Path for every flow. From the experiments based on real network and global typical topology, we conclude that CATD has four advantages: (1) because of the optimized algorithm, CATD realizes congestion avoidance rapidly; (2) CATD makes full use of the other links so as to enhance the bandwidth utilization of the whole network; (3) the constructed CA-Path is obviously shorter than the path of the other mechanism; (4) the calculation of CA-Path employs those links which will not be dangerous or congested after bypassing so CATD will not lead to more congestion.

This paper has four main contributions:

(1) CATD mechanism constructs CA-Path which effectively bypasses the traffic of dangerous link and avoids congestion, and it keeps the forwarding continuous.

(2) The optimization of CA-Path based on two-dimensional forwarding and the innovative consideration about the added entries reduces redundant deployment of forwarding entries, promotes the efficiency of bypassing and improves the network availability.

(3) CATD is carried out in the upstream node which detects congestion danger, which means the mechanism is active even when the congestion danger is short-lived.

(4) CATD subdivides forwarding by source and destination node, increases the granularity of avoidance, and satisfies the demand of traffic reduction.

The paper is organized as follows. In Section 2, we present the related work and background. CATD mechanism including the calculation and optimization of CA-Path is presented in Section 3. Section 4 introduces the strategy to select congestion avoidance flow. The performance evaluation is given in Section 5. Section 6 is the summary of the whole paper.

## 2. Related work

To solve the problem of frequent network congestion, researchers put forward different mechanisms to implement congestion avoidance or congestion control. Loop Free Alternates mechanism (LFA) [5] is a fast failure protected congestion control mechanism built on the network layer, and its core work is to quickly respond to link or node failure by calculating the loop free back-up path automatically. The advantage of LFA is that its calculation process is conducted without any support from other routers. Once the link failure occurs, the path can be immediately switched, which has no possibility of looping. However, the protection coverage of LFA is greatly affected by the network topology. For this problem, G. Retvari et al. [6,7] have extended the research of LFA and presented the mechanism of increasing the protection coverage of LFA. In their first paper [6], considering the limitation of the formula used in LFA technology for the next-hop, they presented a mechanism to maximize the protection coverage of LFA by optimizing IGP (Interior Gateway Protocol) link cost. Moreover [7], they have used the greedy algorithm to find the topology in which the loop-free next-hop must exist. And they also put forward the strategy of adding 2–3 links in topology to increase the protection coverage of LFA. But these methods rely on pre-computing and path backups, which are inconvenient to solve the problem of unpredictable congestion. An active failure insensitive routing (FIR) method based on local rerouting has been proposed [8], which uses interface-specific forwarding to prepare for the occurrence of failure, suppresses the global announcement when failure occurs, and triggers backward routing table to reroute. FIR also introduces the concept of critical links to avoid forwarding back to the fault-prone links. Z. Zhong et al. [9] have proposed a mechanism called failure inferencing based fast rerouting (FIFR), which uses the forwarding table in each linecard to deal with the emergent failure, announces only when the failure continues, and insures the high service availability without changing the original forwarding paradigm.

The development of tunneling technology provides a new means of congestion avoidance. IP fast reroute using Tunnels [10] has been put forward, which can be carried out by triggering the tunnel node near the congestion occurrence point. At the same time, the selection of the tunnel endpoint is restricted to avoid the occurrence of transmission loop. Optimized method rLFA (Remote Loop Free Alternates) [11] is to solve the problem that LFA cannot protect all the failed links, which backups multiple tunnel endpoints for link to implement congestion control. Y. Yang et al. [12] have presented an IP fast reroute mechanism based on tunneling, called Fast Tunnel Selection (FTS). This algorithm optimizes the computational process of the tunnel endpoint and finds out the tunnel endpoint before the accomplishment of a full shortest path tree (SPT) calculation, effectively reducing the calculation burden. L. Pan et al. [13,14] have optimized Tunnels and proposed Tunnel-AT. In Tunnel-AT, when failure occurs, the failure-adjacent node will transmit the packet using the node-failure backup paths or the link-failure backup paths. Furthermore, the computational cost of this strategy is lower than one full SPF (Shortest Path First) calculation because it is based on incremental SPF algorithm and the concept of Attaching Tree. However, congestion avoidance by tunneling will add an extra field to the message, and the path calculated by tunneling is only a simple detour path, which will repeat the previous transmission path.

Conventional mechanisms usually bypass the flow to one specific destination to avoid the congestion of single link. However, if the flow to a certain point is all shunted, on the one hand, it may not realize the congestion avoidance, on the other hand, it may make the link completely idle. If the flow can be subdivided, the congestion avoidance can get higher granularity of bypassing. M. Xu et al. [15,16] have defined and introduced two-dimensional IP routing, and in order to solve the compatibility problem of conventional route device, a new forwarding table structure was designed, including simple Policy Routing Protocol. Because of the difficulty of its compatibility, they have proposed the incremental deployment design for two-dimensional IP routing in the later research. Two-dimensional routing which marks forwarding entry by source and destination prefix enables the router to forward packets to the same destination address along different paths according to different source address, and provides theoretical support for multipath transmission of congestion avoidance.

Multipath routing is an effective strategy to realize congestion avoidance and load balance. Equal-cost multipaths (ECMP) [17]

evenly distributes traffic into multiple equal cost paths, but these paths are statically fixed and cannot react immediately to the network congestion state. To avoid the contention of bandwidth and increase security, disjoint multipath is proposed. N. Taft-Plotkin et al. [18] have proposed a mechanism of calculating the maximum disjoint path in advance, which is used to choose the path statically. S. Nelakuditi et al. [19] have proposed the proportion routing of widest disjoint path (WDP), and the traffic is proportional in multiple widest disjoint path. According to the proposed proportion routing, traffic is distributed proportionally to several better paths, rather than solely transmitted along the best path. In order to increase the utilization of bandwidth, A. Gopalan et al. [20] have put forward the algorithm which constructs three independent spanning trees with random destination nodes as root. In these three trees, paths from random node to the root are disjoint. And the path selection of the packet is based on mod 3 calculation value of the destination port number. H. Su [21] has proposed IP Local Fast-Reroute (IPLFRR). In this strategy, when a node or a link fails, adjacent node will trigger local rerouting and sends packet to the back-up next-hop. The back-up next-hop has been chosen in advance, and the transmission delay can be limited to a few milliseconds. K. Xu et al. [22] have proposed LBMP to achieve Internet traffic management using a logarithm-barrier-based approach, which increases the multipath utility by deploying logarithm barriers at the constraint boundary. However, these multipath transmission methods are still not real-time enough for congestion avoidance, and the response after congestions or fails will inevitably cause transmission delay.

During the whole routing process, calculating and storing the backup path will cause computing and storage burden, and larger scale backup calculation will be carried out if we need to protect all the links. Most congestion avoidance mechanisms use back-up method to deal with congestion, which is undoubtedly costly. Our mechanism performs local bypassing calculation before the congestion occurs, and no path backup is required in CATD, so CATD is a real-time mechanism.

## 3. Congestion avoidance strategy

Existing intra-domain routing usually uses the shortest path to transmit packet, which sometimes leads to link congestion. For example, in Fig. 1(a), the link cost is given, and three flows from A, B and C are sent to node F through the link $e(D, F)$ ($e(N_1, N_2)$ expresses the directed link between the node $N_1$ and the node $N_2$) result in $e(D, F)$ being congestion danger. We hope that part of the traffic in dangerous link can reach the destination node via suboptimal path, for example in Fig. 1(b) the blue flow (C-D-F) avoids $e(D, F)$ and transmits along the suboptimal path as C-D-E-F.

In this paper, a Congestion Avoidance transmission mechanism based on Two-Dimensional forwarding (CATD) is designed to avoid single link congestion, to bypass some traffic in the congestion dangerous link. In CATD, bandwidth threshold for link is set to judge whether a link is in the danger of congestion, and it is defined by the network administrator according to the demand of the network. The demand for high link utilization needs a high threshold, and the demand for safety corresponds to a low threshold. The CATD is mainly designed for routers in single autonomous domain, and based on the most usual intra-domain routing protocol OSPF (Open Shortest Path First) [23]. What is more, this mechanism mainly deals with the network congestion avoidance at single direction of single link. As for the congestion of multi-links, it can be extended and realized on the basis of CATD scheme, which belongs to our future research. In some cases, directly bypassing some flows to other paths may lead to more congestions in some links which have their own

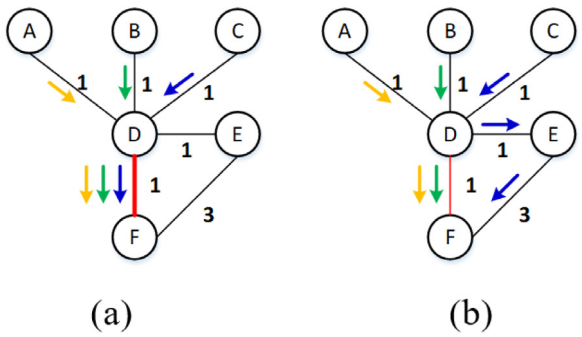


**Fig. 1.** Example of congestion avoidance transmission. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

loads before or after the bypassing. This kind of link is in the "danger" of congestion as well. The SDN server learns all the link states in one time period, which could help filter dangerous link and use a safe topology (the topology which has filtered out all the dangerous link and potential dangerous link) in the calculation of bypassing path. This measure protects CATD from leading more congestions in a certain extent, and reduces the heavy load on the key links while making use of some other free links. This process is good for the load balance of the whole network. After that, if the congestion danger happens again, CATD can be available as well. Without such congestion avoidance mechanism, the whole network may become saturation after a certain key link is overloaded. While with this mechanism, after continuous load balancing, the entire network can carry more traffic. For the long-term effects of the network, CATD improves the bandwidth utilization of the whole network.

The starting node of the link can receive the exception directly, so the avoidance process is triggered by the upstream node of the dangerous link. In order to minimize the impact on other network nodes, the computation of the Congestion Avoidance Path (CA-Path) is implemented immediately in the upstream node with the assistance of SDN server.

CATD solves the congestion problem by bypassing transmission which uses suboptimal path to transmit part of the traffic. Multi-path routing is generally used to balance the traffic of the whole network, while CATD is a dynamic adjustment method when the link is "dangerous".

The CATD implements the congestion avoidance by two-dimensional transmission, which divert the flow based on source and destination address, therefore it is unnecessary to consider the conflict between the CA-Path and raw one-dimensional path. Nodes along the CA-Path will forward the Congestion Avoidance Flow (CA-Flow) according to two-dimensional entries, and other flows will be transmitted along their original shortest path. In Fig. 1, though all flows are sent to the node F, the node D forwards the blue flow (C-D-E-F) by two-dimensional routing, at the same time the orange flow (A-D-F) and green flow (B-D-F) remain their original one-dimensional transmission. Most of the current congestion avoidance mechanisms directly switch the path of some traffic to specific destinations, but our mechanism can divide all the traffic more meticulously based on their sources and destinations. Since flow is the operating unit of CATD, a detailed division of flow is also conducive to finding an appropriate combination of flows for bypassing.

The notations used in this paper are summaried in Table 1.

### 3.1. Congestion avoidance path

**Definition 1.** Suppose that $S$ and $D$ are two edge routers in intra-domain network, $T\_N_{S\to D}$ is the flow between directed node

**Table 1**
Summary on notations.

| Notation | Meaning |
|---|---|
| $N_{up}$ | The upstream node of the dangerous link |
| $N_{down}$ | The downstream node of the dangerous link |
| $S$ | The entrance point of a specific traffic in domain |
| $D$ | The exit point of a specific traffic in domain |
| $T\_N_{S\to D}$ | The directed traffic from the node S to the node D |
| $PATH_{former}$ | Ordered nodes in CA-Path before dangerous link |
| $PATH_{later}$ | Ordered nodes in CA-Path after node $N_{up}$ |
| $S\_N_{dupl}$ | Intersection of $PATH_{former}$ and $PATH_{later}$ |
| $ND_{modify}$ | The node which modifies its next-hop after using CA-Path |
| $S\_ND_{modify}(S, D)$ | The collection of $ND_{modify}$ for $T\_N_{S\to D}$ |
| $T\_P(PFX_1, PFX_2)$ | The flow from $PFX_1$ to $PFX_2$ |
| $S\_PFX(P)$ | The collection of prefixes connected with router $P$ |
| $NUM_{T\_P(S,D)}$ | The number of flows based on prefixes between node $S$ and $D$ |
| $N\_TD_{S\to D}$ | The total number of added entries for $T\_N_{S\to D}$ |
| $B_{realtime}$ | The real-time bandwidth |
| $B_{threshold1}$ | The warning bandwidth threshold |
| $B_{threshold2}$ | The acceptable bandwidth threshold |
| $B_{reduce}$ | The total reduced bandwidth |
| $\tau_i$ | One flow passing the dangerous link |
| $s\Gamma$ | The collection of congestion avoidance flow |

pair$\langle S, D\rangle$, $S$ is the source of the flow $T\_N_{S\to D}$, and $D$ is the destination of $T\_N_{S\to D}$.

It is necessary to explain the difference between "traffic" and "flow". "Traffic" abstractly denotes the message not specific to its source and destination, whereas, "flow" means the specific message which has certain source node and destination node.

In the routing algorithm of OSPF, flow with the same source and destination is usually transmitted along the same path. Apparently, every calculation of CA-Path will introduce calculation load to the trigger node. Therefore, in order to decrease the number of CA-Path and distinguish the flows meticulously, this paper implements the congestion avoidance based on flow ($T\_N_{S\to D}$), which means the bypassing is aimed at flow with the node pair of same source and destination, but not solely aimed at traffic to specific destination node.

The bypassing processes of CATD are as follows. Suppose $e(N_{up}, N_{down})$ is a directed congestion dangerous link, $T\_N_{S\to D}$ is the selected flow which always passes the link $e$, and CATD will bypass $T\_N_{S\to D}$ to another path which does not include $e$. Suppose the raw path along which $T\_N_{S\to D}$ is transmitted before the congestion occurs:

$$S, N_1, N_2, \ldots, N_i, N_{up}, N_{down}, N_{i+1}, N_{i+2}, \ldots, N_k, D$$

For CATD mechanism, $N_{up}$ is the trigger to calculate the CA-Path, $N_{up}$ will calculate its shortest path to the node D based on the safe topology (provided by the SDN server), from which the dangerous link and all the links tending to be dangerous are taken away:

$$N_{up}, N'_{i+1}, N'_{i+2}, \ldots, N'_{i+j}, D$$

The CA-Path of $T\_N_{S\to D}$ is:

$$S, N_1, N_2, \ldots, N_i, N_{up}, N'_{i+1}, N'_{i+2}, \ldots, N'_{i+j}, D$$

### 3.2. Optimized CA-path

Apparently, the later calculation of SPF is different from the original one because the network topology has changed (not including the dangerous link), so the forwarding loop may appear in the CA-path. In the described "loop", the later path after $N_{up}$ may transmit the packet to a node which has been used before, and maybe there are multiple nodes which are used twice in the whole transmitting path. The transmission of Congestion Avoidance Flow (CA-Flow) is based on two-dimensional transmission, therefore it may seem to be a "LOOP" (repeated node in CA-Path), which will not lead to a real infinite loop transmission, but it will still add redundant nodes into CA-Path and introduce redundant storage of routing item. As a result, the CA-Path should be optimized by removing extra nodes to increase the efficiency of deployment and transmission. Repeated nodes need to be found in these two collections:

(1) the ordered nodes collection in the CA-Path before the node $N_{up}$, and $N_0$ is the source node $S$;

$$PATH_{former} = \{N_0, N_1, N_2, \ldots, N_i\}$$

(2) the ordered nodes collection in the later path of the CA-Path after the node $N_{up}$ without the node $D$.

$$PATH_{later} = \{N'_{i+1}, N'_{i+2}, \ldots, N'_{i+j}\}$$

**Deduction 1**: The intersection of two collections above is expressed as:

$$S\_N_{dupl} = PATH_{former} \bigcap PATH_{later}$$

And if: $S\_N_{dupl} \neq \emptyset$, it indicates that there is a loop in the CA-Path.

Transmission loop in the CA-Path means there are repeated nodes in the path, therefore the final optimized CA-Path should remove all redundant nodes to ensure that there exists no loop.

Suppose that:

$$PATH'_{former} = PATH_{former} - S\_N_{dupl}$$
$$PATH'_{later} = PATH_{later} - S\_N_{dupl}$$

$N_k$ is the first repeated node in $PATH_{former}$, it should satisfy the Formula (1).

$$k = min\{f \mid N_f = N'_l, N_f \in PATH_{former}, N'_l \in PATH_{later}, 0 \le f \le i, i+1 \le l \le i+j\} \quad (1)$$

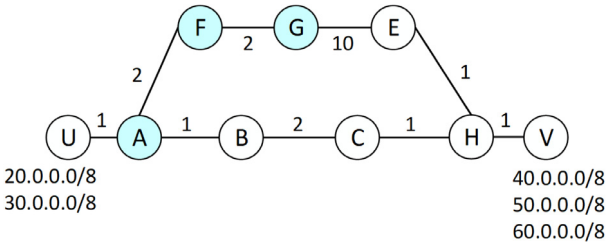


**Fig. 2.** The optimized CA-path topological example.

The final optimized CA-Path can be presented as the Formula (2).

$$\begin{cases} \{PATH_{former}, N_{up}, PATH_{later}, D\} & PATH_{former} \bigcap PATH_{later} = \emptyset \\ \{PATH'_{former}, N_k, PATH'_{later}, D\} & PATH_{former} \bigcap PATH_{later} \neq \emptyset \end{cases} \tag{2}$$

As shown in Fig. 2, when link $e(A, B)$ is about to congest, for flow $T\_N_{U\to V}$, the CA-Path is {U, A, F, G, E, H, V} which is calculated by node A. $PATH_{former} = \{U\}$, $PATH_{later} = \{F, G, E, H\}$, so the $S\_N_{dupl} = PATH_{former} \bigcap PATH_{later} = \emptyset$, and there is no loop in CA-Path, and the path remains unchanged:

$$\{U, A, F, G, E, H, V\}$$

When link $e(C, H)$ is about to congest, for flow $T\_N_{U\to V}$, the CA-Path is {U, A, B, C, B, A, F, G, E, H, V} which is calculated by node C. $PATH_{former} = \{U, A, B\}$, $PATH_{later} = \{B, A, F, G, E, H\}$, at this time:

$$S\_N_{dupl} = PATH_{former} \bigcap PATH_{later} = \{A, B\} \neq \emptyset.$$

$$PATH'_{former} = PATH_{former} - S\_N_{dupl} = \{U\}$$

$$PATH'_{later} = PATH_{laetr} - S\_N_{dupl} = \{F, G, E, H\}$$

Apparently, the first repeated node $N_k$ is A, therefore the optimized CA-Path is:

$$\{PATH'_{former}, N_k, PATH'_{later}, D\} = \{U, A, F, G, E, H, V\}$$

### 3.3. Two-dimensional forwarding entries of CA-path

For the specific flow $T\_N_{S\to D}$, the original transmission paths of some nodes in CA-Path are changed. In this paper, CATD realizes different forwarding for different flows to the same destination by two-dimensional forwarding. The next hop of one node to node V in the Raw Path depends on the shortest path of this node to node V, and most of the original transmission did not use the link $e(G, E)$, because of its cost is 10. For example, there are two paths for F to V ($F \to G \to E \to H \to V$ and $F \to A \to B \to C \to H \to V$), but the cost of first path is more than that of the second path, as the costs marked beside every links, so in the Raw Path, the next hop of F to V is A. As for the node E, the next hop of E is H because of the cost is far less than the cost when the next hop is G. In Fig. 2, when link $e(C, H)$ is about to congest, node C will reroute the transmission path for $T\_N_{U\to V}$, and the CA-Path is: {U, A, F, G, E, H, V}. In this example, there are three nodes (A, F, G) which change their next-hop to the destination node V, as shown in Table 2. For example, the one-dimensional forwarding next-hop to node V (Raw Path) of node F is A but in CA-Path, for the flow $T\_N_{U\to V}$, the next-hop of node F is G.

**Definition 2.** For the flow $T\_N_{S\to D}$, if there is one node in CA-Path whose transmission path is different from its original path to destination D, this node is called Routing Modified Node ($ND_{modify}$), and the collection of these nodes for $T\_N_{S\to D}$ is called $S\_ND_{modify}(S, D)$, for example, the $S\_ND_{modify}(U, V)$ is {A, F, G} in Fig. 2.

**Table 2**
The Next-hop of each node in CA-Path.

| The nodes in CA-Path | | U | A | F | G | E | H | V |
|---|---|---|---|---|---|---|---|---|
| Next-hop for V | Raw Path | A | B | A | F | H | V | – |
| | CA-Path | A | F | G | E | H | V | – |

**Definition 3.** Suppose $PFX_1$, $PFX_2$ are two IP prefixes in the network (respectively connected with different edge router nodes), $T\_P(PFX_1, PFX_2)$ is the flow between these two prefixes.

Every AP router in the intra-domain network is connected with at least one IP prefix. Although the best path between the same source node and destination node in AS is identical, the final forwarding table in the router is still deployed according to IP prefix. It means that the number of IP prefixes between two edge nodes will change the number of forwarding entries in the router along the path.

Suppose that the collection of prefixes connected with router $P$ is $S\_PFX(P)$, therefore the prefix pair transmitted through the node pair $\langle S, D\rangle$ is the Cartesian product of $S\_PFX(S)$ and $S\_PFX(D)$. The number of all the flow based on prefixes between edge node $S$ and $D$ can be presented as the Formula (3):

$$NUM_{T\_P}(S, D) = \mid S\_PFX(S) \mid * \mid S\_PFX(D) \mid \tag{3}$$

In Fig. 2, router U connects with two IP subnets: 20.0.0.0/8 and 30.0.0.0/8. At the same time, router V connects with three IP subnets: 40.0.0.0/8, 50.0.0.0/8 and 60.0.0.0/8. In that way, the number of flows based on prefixes which are transmitted along $T\_N_{U\to V}$ will be 2*3 = 6.

20.0.0.0/8 → 40.0.0.0/8  20.0.0.0/8 → 50.0.0.0/8
20.0.0.0/8 → 60.0.0.0/8  30.0.0.0/8 → 40.0.0.0/8
30.0.0.0/8 → 50.0.0.0/8  30.0.0.0/8 → 60.0.0.0/8

The congestion avoidance is implemented by two-dimensional forwarding. If the flow $T\_N_{U\to V}$ which is based on nodes is determined to construct CA-Path, the number of two-dimensional forwarding entries added into each $ND_{modify}$ equals to the number of flows based on prefixes which are transmitted from U to V, that is $NUM_{T\_P}(U, V)$.

As for the flow $T\_N_{S\to D}$, the Definition 2 describes the collection of nodes in CA-Path which change their route: $S\_ND_{modify}(S, D)$. Therefore, the total number of two-dimensional forwarding entries added in CA-Path will be presented as the Formula (4):

$$N\_TD_{S\to D} = \mid S\_ND_{modify}(S, D) \mid * NUM_{T\_P}(S, D) \tag{4}$$

In Fig. 2, the number of $ND_{modify}$ in CA-Path for $T\_N_{U\to V}$ is 3, so the $N\_TD_{U\to V}$ is 3*6 = 18.

### 3.4. The continuity of routing modified node

Considering the configuration of $ND_{modify}$ in CA-Path, according to the SPF algorithm, the shortest path of source node to destination node is bound to cover the paths of every node in this path to the same destination node, then comes the deduction.

**Deduction 2**: For the flow $T\_N_{S\to D}$, if a node $P$ in CA-Path whose next-hop has no difference from its original next-hop to $D$, then every node after node $P$ in CA-Path will not be a $ND_{modify}$.

**Deduction 2** can be simply described as: in the ordered node collection of CA-Path, the Routing Modified Node always comes continually.

According to the **Deduction 2**, the deployment of two-dimensional forwarding entry can be further simplified. The upstream node of dangerous link calculates the number of $ND_{modify}$ and the number of $N\_TD_{S\to D}$ by Algorithm 1.

**Algorithm 1:** Calculation of $N\_TD_{S\to D}$

**Data**: $Topology, S, N_{up}, N_{down}, D$
**Result**: $N\_TD_{S\to D}$
calculate and optimize the CA-Path for the $T\_N_{S\to D}$;
$count = 0$;
$node = N_k$;
**while** *node in CA-Path* **do**
  **if** $node.nexthop$!=$ori\_nexthop(node, D)$ **then**
    //$ori\_nexthop(node, D)$ returns the next-hop of *node* when $D$ is the destination
    $count++$;
  **else**
    BREAK;
  **end**
  $node++$;
**end**
$N\_TD_{S\to D} = count * prefix(S) * prefix(D)$;
RETURN $N\_TD_{S\to D}$;

### 3.5. The construction of CA-path

Each congestion avoidance flow $T\_N_{S\to D}$ has a CA-Path. Obviously, not every node in CA-Path needs to add forwarding entry for $T\_N_{S\to D}$. According to the Formula (2) and Algorithm 1, the nodes in CA-Path which need to add entry are several continuous nodes starting from $N_{up}$ or $N_k$. Suppose that the source IP prefixes connected with $T\_N_{S\to D}$ (the IP prefixes connected with $S$) is $PFX\_S_{1\to snum}$, and the destination IP prefixes (the IP prefixes connected with $D$) is $PFX\_D_{1\to dnum}$.

The construction of CA-Path is triggered and calculated by the upstream node of dangerous link $N_{up}$, implemented by the SDN node of the area. And with the help of the SDN server, the calculation of CA-Path will proceed on a safe topology which has filtered out those links tending to be dangerous after bypassing. So it is unnecessary to consider the extra congestion result from CATD. To reduce the transmitting burden between the SDN server and $N_{up}$, the filtering process eliminates links which will be dangerous after loading all the reduced bandwidth of bypassing. For example, as shown in Fig. 3, the number on every link represents the load of each link. Suppose that when the load of a link is more than 80M, it will be recognized as dangerous, and the bypassing process needs to make the load no more than 60M. In Fig. 3(a), the link $e(C, D)$ loads 80M which has been recognized as dangerous link and needs to bypass some flows with the bandwidth of 20M. For the other links in this topology, such as $e(A, F)$, $e(H, D)$ and $e(B, I)$, they will be dangerous after loading 20M bandwidth. So they need to be filtered in the calculation of CA-Path as shown in Fig. 3(b).

The procedures are as follows:

Step 1, $N_{up}$ calculates the CA-Path, which is the shortest path from $N_{up}$ to $D$ in topology without link $(N_{up}, N_{down})$ and those links which has been filtered.

Step 2, $N_{up}$ optimizes the CA-Path and calculates the collection of ordered nodes which are needed to add forwarding entry:

$$S\_ND_{modify}(S, D) = \{ND\_M_1, ND\_M_2, \ldots, ND\_M_m\}$$

The order of nodes in the collection is the same as the order of these in CA-Path.

Step 3, $N_{up}$ sends the message ($MSG_{TD}$) about the two-dimensional forwarding strategy to nodes in $S\_ND_{modify}(S, D)$. The $MSG_{TD}$ includes: (1) source prefix; (2) destination prefix; (3) next-hop. The sequence of announcement is the reversed order of the collection (from $ND\_M_m$ to $ND\_M_1$), which insures the validity of the transmission path. When $ND\_M_1$ transmits flow by two-dimensional forwarding entry, subsequent path has been constructed successfully.

Step 4, the nodes in $S\_ND_{modify}(S, D)$ add two-dimensional forwarding entry according to received $MSG_{TD}$ : ($PFX\_S_{1\to snum}$, $PFX\_D_{1\to dnum}$, $next - hop$), and the total number of entries is: $snum * dnum$.

As shown in Fig. 2, when link $e(A, B)$ is about to congest, $S\_ND_{modify}(U, V) = \{A, F, G\}$, according to the IP prefixes connected with node U and V ($PFX\_U_{1\to 2}$, $PFX\_V_{1\to 3}$), every node needs to add 6 forwarding entries. Then the message $MSG_{TD}$ is sent backwards to these three nodes as the order of G-F-A, and the total number of entries added is $3 * 6 = 18$.

## 4. The selection of congestion avoidance flow

### 4.1. Basic selection rule of congestion avoidance flow

The CATD traffic control strategy for congestion avoidance sets thresholds for key links, and makes part of its traffic bypass the link when the load of the link exceeds the threshold. Suppose that the bandwidth represents the size of all the traffic in question, for example, the load bandwidth of specific link is the size of all the flows it loads regardless of the kind of the flow. Suppose the link warning bandwidth threshold is $B_{threshold1}$, the acceptable (safe) bandwidth threshold is $B_{threshold2}$ ($B_{threshold1} \succ B_{threshold2}$), the real-time bandwidth is $B_{realtime}$, and the adjusting flow bandwidth is $B_{reduce}$. Then the rule of congestion avoidance can be described as: when the bandwidth of detected link satisfies $B_{realtime} \geq B_{threshold1}$, then it is necessary to make part of the traffic with total bandwidth of $B_{reduce}$ transmits along CA-Path, and $B_{reduce}$ satisfies $B_{realtime}$ - $B_{reduce} \leq B_{threshold2}$.

Suppose that the collection of all the flow ($T\_N$) which passes the dangerous link $e(N_{up}, N_{down})$ is $\Gamma = \{\tau_1, \tau_2, \ldots \tau_n\}$, and the specific flow $\tau_i$ always passes the intra-domain network through the same source router ($S_i$) and destination router ($D_i$). Moreover, the source router and the destination router of two random flows $\tau_i$ and $\tau_j$ cannot be identical, which can be described as $(S_i \neq S_j) \| (D_i \neq D_j)$. Suppose the bandwidth used by flow $\tau_i$ is $B_{\tau i}$, there must be a minimum subset $s\Gamma$ of $\Gamma$ whose total bandwidth satisfies the rule of congestion avoidance. The $s\Gamma$ is called Congestion Avoidance Flow Collection which satisfies the Formula (5). Bypassing all the flow in one $s\Gamma$ and transmitting them along CA-Path will realize the target of congestion avoidance.

$$\begin{cases} s\Gamma = \{\tau_i \mid \tau_i \in \Gamma, \sum B_{\tau_i} \geq B_{realtime} - B_{threshold2}\} \\ \forall \tau_j \in s\Gamma, \sum B_{\tau_i} - B_{\tau_j} \leq B_{realtime} - B_{threshold2} \end{cases} \tag{5}$$

Apparently, there may be several $s\Gamma$ which satisfy the demands above. Under this circumstance, the total two-dimensional forwarding entries added in each collection should be considered when selecting a proper $s\Gamma$. Suppose that $\tau_i$ is a flow in $s\Gamma$, the source node and the destination node of $\tau_i$ is $S_i$ and $D_i$, and every $\tau_i$ has its CA-Path. According to the Formula (4), the number of two-dimensional forwarding entries added in CA-Path of $\tau_i$ can be presented as Formula(6).

$$N\_TD_{\tau_i} = N\_TD_{S_i\to D_i} = \mid S\_ND_{modify}(S_i, D_i) \mid * NUM_{T\_P}(S_i, D_i) \tag{6}$$

Therefore, the number of two-dimensional forwarding entries added in the $s\Gamma$ can be presented as the Formula (7):

$$N\_TD(s\Gamma) = \sum N\_TD_{\tau_i}, \tau_i \in s\Gamma \tag{7}$$

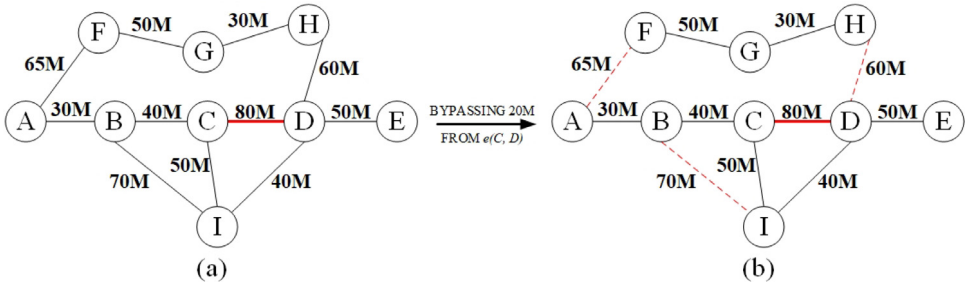


**Fig. 3.** Example of link filtering.

**Table 3**
Detailed data of Fig. 4.

| $T\_N$ | Source/Destination | Bandwidth | CA-Path | $S\_ND_{modify}$ | $NUM_{T\_P}$ | $N\_TD_{\tau i}$ |
|---|---|---|---|---|---|---|
| $\tau_1$ | A/H | 100 M | A-**E**-**G**-F-H | \|{E,G}\|=2 | 16 | 32 |
| $\tau_2$ | A/I | 60 M | A-**E**-C-I | \|{E}\|=1 | 8 | 8 |
| $\tau_3$ | B/H | 40 M | B-**E**-**G**-F-H | \|{E,G}\|=2 | 4 | 8 |
| $\tau_4$ | B/I | 30 M | B-**E**-C-I | \|{E}\|=1 | 2 | 2 |
| $\tau_5$ | C/H | 30 M | **C**-I-F-H | \|{C}\|=1 | 8 | 8 |

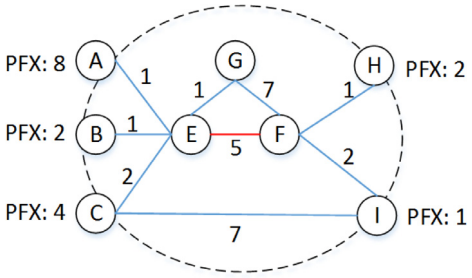


**Fig. 4.** Example of CA-flow selection.

The $s\Gamma$ with the least $N\_TD(s\Gamma)$ will be selected to implement congestion avoidance, because it has the minimal impact on router storage. The description above of selection strategy can be represented as the mathematical model in Formula (8), where the $\Gamma$ means the ordered collection of flows in the dangerous link, and the $x_i$ is the flag value to express whether $\tau_i$ is chosen to be bypassed, $N\_TD_{\tau_i}$ is the number of entries added in the CA-Path of $\tau_i$.

$$o.f.\ min \sum_{\tau_i \in \Gamma} x_i \cdot N\_TD_{\tau_i} \tag{8}$$

$$s.t.\ \sum x_i \cdot B_{\tau_i} \geq B_{reduce}$$
$$\forall B_{\tau_i} \geq 0$$
$$\forall x_i \in \{0, 1\}$$

For example, as shown in Fig. 4, the cost of each link in the AS topology is given. There are 5 edge nodes: A, B, C, H, I, and the number of subnets connected with each node is: 8, 2, 4, 2, 1. The link $e(E, F)$ is going to congest, and it is necessary to bypass part of the traffic with bandwidth of 100M. All the flow passing the dangerous link based on nodes and its CA-Path are shown in Table 3. $|S\_ND_{modify}|$ describes the nodes and the number of nodes which need to add forwarding entry, $NUM\_T_P$ is the number of flows based on prefixes which are transmitted by this $T\_N$, and $N\_TD_{\tau i}$ is the number of forwarding entries added in this CA-Path.

According to the flow selection strategy in the Formula (5), there are multiple flow collections which satisfy the demand of congestion avoidance (suppose that the $B_{reduce}$ is not less than 100M). Then the collection with the least $N\_TD(s\Gamma)$ will be selected to implement congestion avoidance and the flows within will be transmitted along CA-Path. After calculation of the $s\Gamma$, there are four sub-collections of $\Gamma$ which satisfy the demand of bypass and their $N\_TD(s\Gamma)$ is shown in Table 4. It is obvious that the collection "$\{\tau_2, \tau_3\}$" will be chosen to bypass the dangerous link, because the number of entries that adds into will be the least.

**Table 4**
Bypassing flow collection in Table 3.

| $s\Gamma$ | Bandwidth | $N\_TD(s\Gamma)$ |
|---|---|---|
| $\{\tau_1\}$ | 100 M | 32 |
| $\{\tau_2, \tau_3\}$ | 100 M | 16 **(selected)** |
| $\{\tau_2, \tau_4, \tau_5\}$ | 120 M | 18 |
| $\{\tau_3, \tau_4, \tau_5\}$ | 100 M | 18 |

According to the selection strategy above, a proper collection of CA-Flow is selected. Then the flows in this collection are going to bypass the dangerous link and transmit along their CA-Path. In order to make sure the paths of these flows can be successfully switched, the two-dimensional forwarding entry should be deployed in $ND_{modify}$ using Algorithm 2. The sequence of deployment is as described in Section 3.5.

**Algorithm 2:** The Deployment of Two-Dimensional Entry

```
Data: Topology, sΓ
Result: The Execution Result
for T_N_{S→D} in sΓ do
    for node = ND_M_m to ND_M_1 do
        for PFX_i in S_PFX(S) do
            for PFX_j in S_PFX(D) do
                Add the Two-Dimensional Forwarding Entry
                in node: < PFX_i, PFX_j, node.nexthop >;
            end
        end
    end
end
RETURN The Execution Result;
```

### 4.2. Aggregation calculation of CA-path

In Table 4, if every $s\Gamma$ is calculated respectively and directly, then some path may be calculated several times. For example, the CA-Path and the $N\_TD_{\tau i}$ of $\tau_2$ will be calculated twice, because $\tau_2$

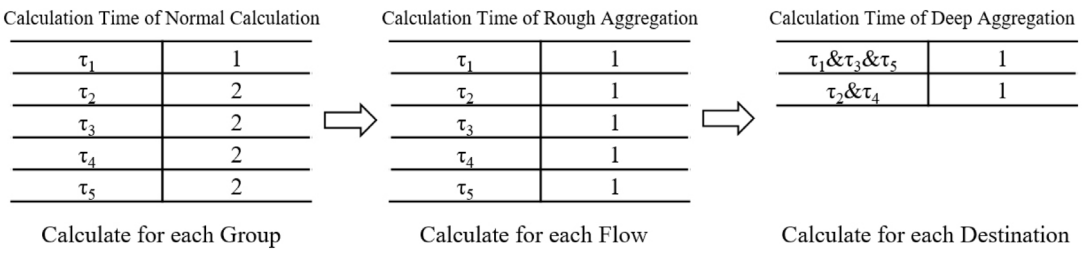


**Fig. 5.** Result of aggregation.

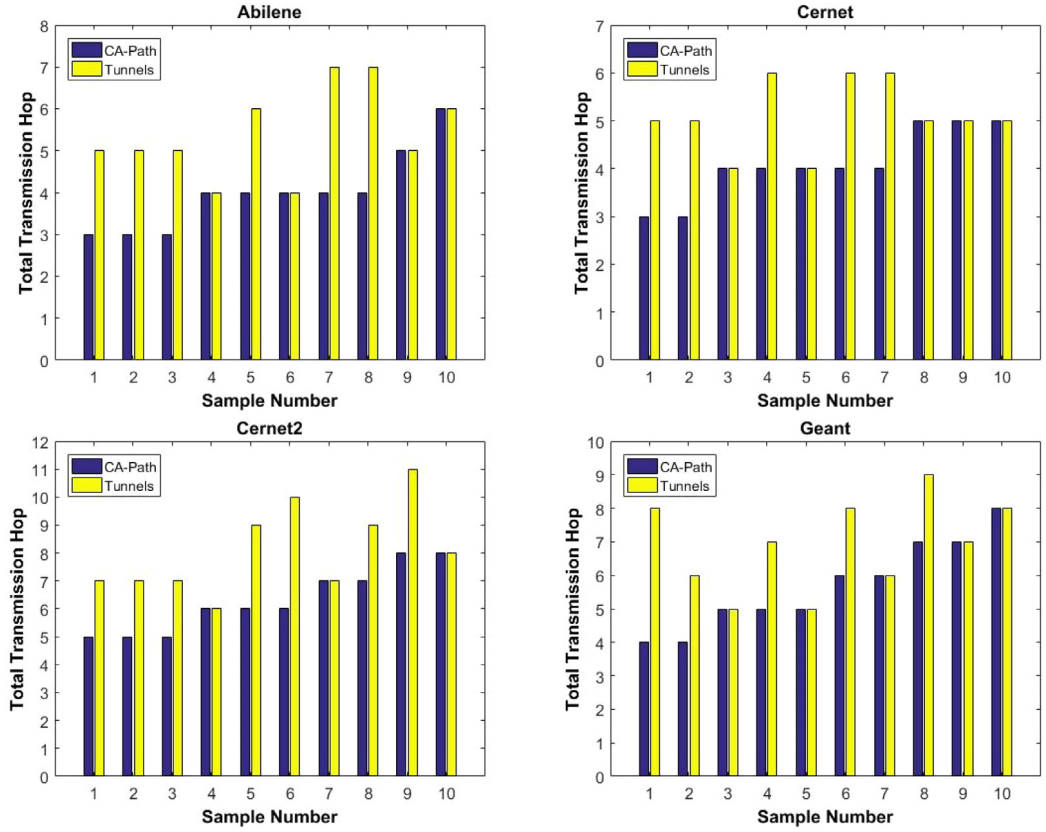


**Fig. 6.** Results comparison of CATD and tunnels.

appears both in the second $s\Gamma$ and third $s\Gamma$. It is bound to add extra calculation burden.

In order to fix this weak point, the group of flows are unpacked at first, after the calculation of each CA-Path and $N\_TD$, recombine these $s\Gamma$, and sum up their total $N\_TD$. In this way, every traffic will be calculated only once, which seems still fussy in calculation for these calculations are ran in only one upstream node.

As in the description above of the CA-Path calculation, the SPF calculation starts from same node ($N_{up}$), and ends in the destination node of that flow ($D$), which comes to a decision that the CA-Path of flow with same $D$ can be calculated aggregately. For example, the destination points of $\tau_1$, $\tau_3$, $\tau_5$ are the node H, so their CA-Path will all be calculated from node E to node H.

**Deduction 3**: For every flow in $e(N_{up}, N_{down})$, CA-Paths of flows with same destination $D$ (the shortest path from $N_{up}$ to $D$) will be the same, so the calculation of SPF for these flows can be merge as one SPF. The result of aggregation is shown in Fig. 5.

The main advantage of aggregated calculation is that it can effectively reduce the heavy burden of calculation for the trigger node. Under the aggregated calculation, the SPF calculation can be compressed in only one time to receive the path from $N_{up}$ to each node $D$, later calculation of $N\_TD_{\tau i}$ is less complex than SPF, so the whole algorithm complexity is $O(n^2)$.

As the $N\_TD_{\tau i}$ of each traffic is known after aggregated calculation, with the mathematical model in Formula (8), the selection can be more simple to solve in any device.

## 5. Performance evaluation

### 5.1. CA-Path optimization

In order to demonstrate the effect of the CA-Path calculation and the optimization algorithm, we realize our own algorithm and congestion avoidance mechanism of Tunnels [10], and these two algorithms are deployed on four real typical network topologies: Abilene [24], CERNET [25], CERNET2[26], and GEANT [27]. We choose ten random samples in each topology, and provide the upstream node and the downstream node of dangerous link and the specific source node and destination node, whose transmission path covers the dangerous link. After the calculation of each algorithm, we analyze the total hop of these two algorithms.

In Fig. 6, the bars of Tunnels are never lower than those of CA-Path, which means the total hops of CA-Path will never be

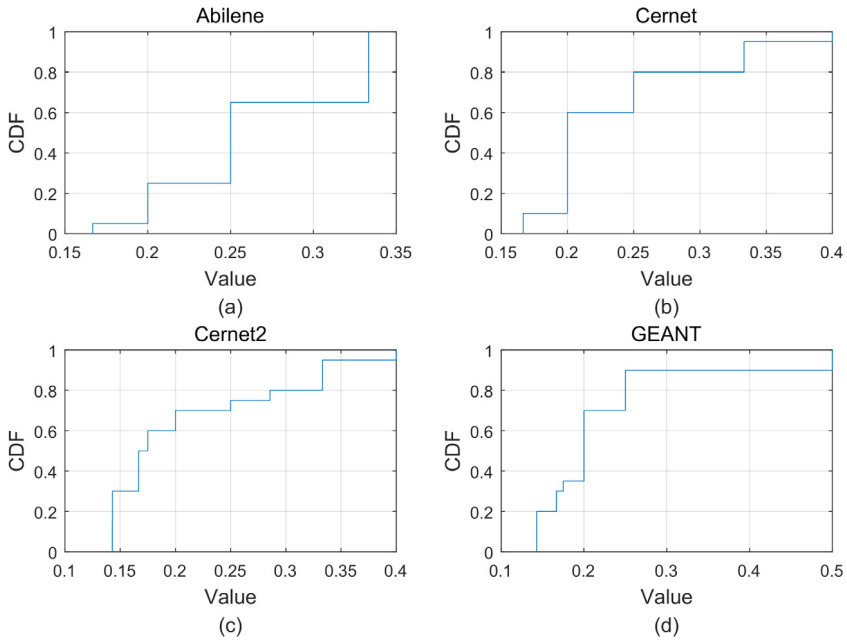


**Fig. 7.** The path deployment ratio of partial deployment.

more than the hops of Tunnels. It is clear that CA-Path uses less hops to finish the transmission, and the optimization effects are remarkable.

### 5.2. Deployment of the CATD

To reflect the partial deployment effect of CATD, we further calculate the number of $ND_{modify}$ in CA-Path after the optimized calculation in the experiment of the last subsection. We select 20 pairs of terminal nodes randomly in the above four real topologies, and select a directed link on its original shortest path as a dangerous link. Then CATD calculates the CA-Path for every pair and the $ND_{modify}$ on the path, and then we obtain the proportion between them. We introduce the conception of Path Deployment Ratio, the proportion of $ND_{modify}$ in CA-Path, to verify the stability of the CATD partial deployment under different topologies and congestion scenarios.

In Fig. 7, cumulative distribution function (CDF) of the Path Deployment Ratio on every topology is shown. From the CDF of the Path Deployment Ratio, we can know the maximum Path Deployment Ratio and the most frequent value for each topology. It is obvious that the Path Deployment Ratio of CATD is steady in different topologies and congestion scenarios. According to the experiment, we can know that CATD only needs to deploy part of the nodes on each CA-Path to complete congestion avoidance, and the deployment ratio is relatively stable, which is from 0.13 to 0.5.

### 5.3. Two-dimensional deployment of CATD

According to the OSPF protocol, we set up a real experiment network based on the BitEngine Router (X86 platform) designed by Tsinghua University. The topology is shown in Fig. 8, and all linkcosts equal to 1. The hardware parameter of router is: CPU, Intel Pentium(R) G2030 3.0 GHz, dual-core; RAM, 4G; Network Interface, 4 GigE ports. The maximum transmission ability of these devices is no more than 600Mbps, and higher traffic will lead to packet loss. The traffic is detected and analyzed by IXIA OptIxia XM12 IP network tester. The detected dangerous link is $e(C, D)$, and the experiment parameter is: $N_{up}$ = C, $N_{down}$ = D,

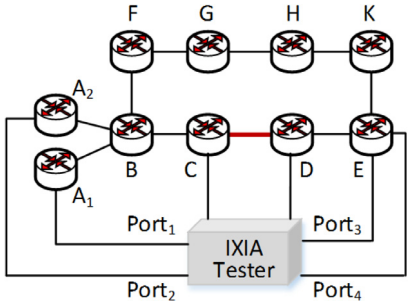


**Fig. 8.** Experiment topology.

$B_{threshold1}$ = 550M, $B_{threshold2}$ = 450M. In order to enhance the effect, we set background traffic ($Traffic_b$) with size of 300Mbps on link $e(C, D)$, which is transmitted from host $H_3$ to $H_4$ passing $e(C, D)$, $H_3$ and $H_4$ are the visual hosts which are used to send data.

During the experiment, the tester sends traffic through $Port_{1-2}$, and receives traffic from $Port_{3-4}$. $Port_1$ continuously sends 80M traffic ($Traffic_1$), this traffic is received by $Port_3$, and the best path is: $A_1 - B - C - D - E$. In addition, $Port_2$ sends traffic to the tester with incremental bandwidth from 0M to 500M ($Traffic_2$), this traffic is received by $Port_4$, and the best path is $A_2 - B - C - D - E$.

The experiment compares CATD with the existing single path transmission (S-Path) which is based on OSPF protocol and has no congestion avoidance action. After detecting and receiving the bandwidth and the unidirectional transmission delay by IXIA Tester, the transmission ability is shown in Fig. 9 (a) and (b). In S-Path model, after $Traffic_2$ reaches 170M, packet loss occurs in this traffic, and the condition becomes severe as time goes on. On the contrary, in CATD, $Traffic_1$ and $Traffic_2$ can always be received normally. When the traffic reaches 550M in $e(C, D)$, $Traffic_2$ switches to another path: $A_1 - B - F - G - H - K - E$. Slight packet loss is the result of path change.

As for the unidirectional transmission delay in Fig. 9 (c) and (d), for $Traffic_1$ in S-Path, the transmission delay increases evidently after congestion due to the influence of $Traffic_2$; on the

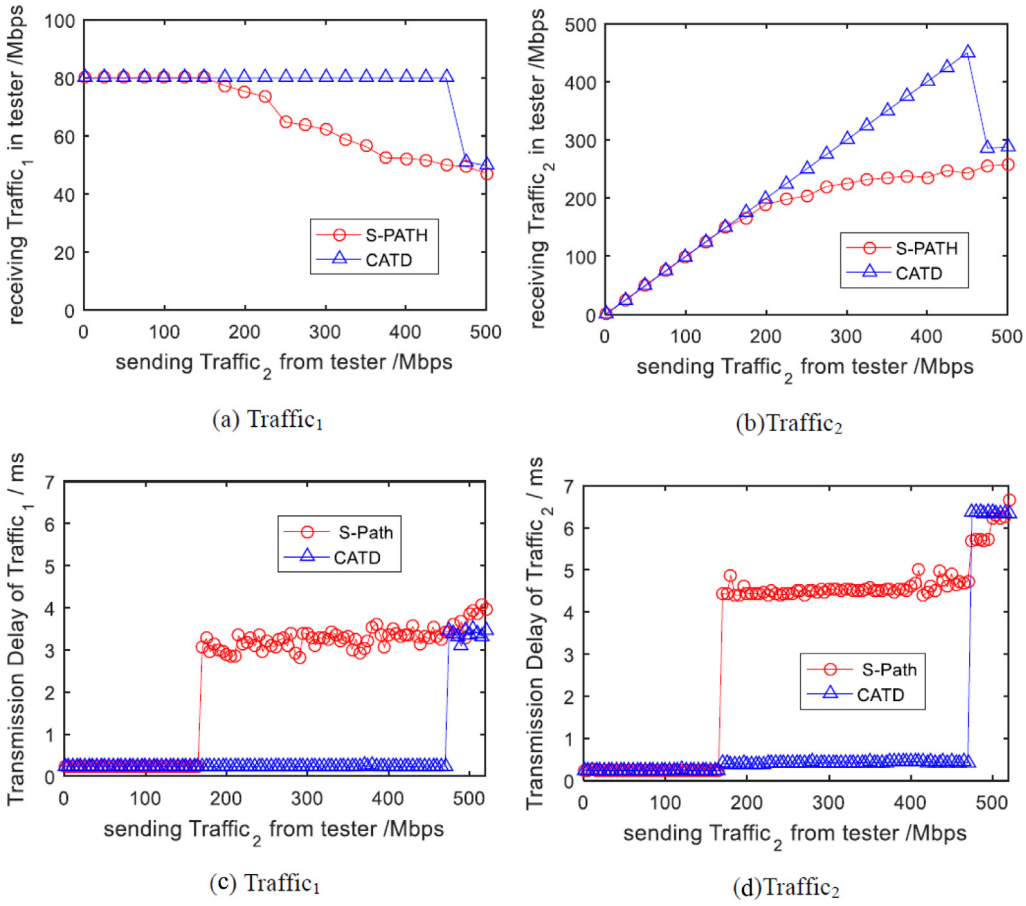


**Fig. 9.** Transmission performance in different models.

contrary, in CATD, the transmission delay remains unchanged because of the implement of congestion avoidance. The general trends of $Traffic_1$ and $Traffic_2$ are roughly similar. The difference is that after congestion avoidance, the delay of $Traffic_2$ is higher, because its path is changed from 5 hops to 7 hops.

We also carry out a comparative experiment on the two-dimensional forwarding performance and one-dimensional forwarding performance on a single device, and both forwarding methods are able to forward messages of more than 600 bytes at a speed of 980Mbps. The experiment shows that the performance of two-dimensional forwarding is basically equal to that of one-dimensional forwarding.

The actual network that deploy CATD only needs to have the following three elements: (1) support two-dimensional forwarding; (2) SDN service nodes monitor link bandwidth; (3) routers support CA-Path calculation and notification. To verify the deployment effects of CATD on large-scale network topologies, we deploy single path transmission, the ECMP transmission and the CATD on the Abilene and GEANT topologies respectively based on devices which satisfy the above conditions. Single path transmission (S-Path, OSPF) has no congestion avoidance function and optimization, ECMP mechanism evenly distributes traffic into multiple equal cost paths. In the experiment, we use 80% as the dangerous threshold and the actual traffic matrix as the initial data. We gradually increase the traffic matrix in proportion and calculate the number of links in the network whose loads have exceeded the congestion threshold.

It is not difficult to see from Fig. 10 that in the gradually increasing flow matrix of CATD, the time when the dangerous links appear is the latest, i.e., the capacity of carrying the flow is the largest. And in the process of the flow matrix increasing gradually, the dangerous links appear least. As can be seen from the figures, when the traffic is becoming large, the dangerous links under the ECMP mechanism are more than that under the S-Path mechanism. This is because, while most links are dangerous and a small number of links with small-occupancy are safe when traffic increases. Neither the S-Path mechanism nor the CATD mechanism will cause these small-occupancy links to be dangerous, but the equalization strategy of ECMP will cause these links to be dangerous. The experimental results show that CATD can be implemented on large-scale topologies and the implementation effect is good.

### 5.4. Bandwidth utilization of CATD

To test the network bandwidth utilization in CATD, we select the real topology Abilene with the traffic matrix from 15:00 to 23:00 a day and GEANT with the 1.5x amplified traffic matrix from 15:00 to 23:00 a day, and two different dangerous thresholds (60% and 80%) are used to balance the traffic in the network in single path transmission, ECMP and CATD. At each hour, the bandwidth utilization of the highest loaded link in the entire network is recorded. Under the same topology and traffic matrix, the traffic carried by the three mechanisms are the same, but the load balance methods of the three mechanisms are different, so the maximum bandwidth occupancy of the link will be different. Therefore, under the same environment, the lower the maximum bandwidth utilization rate of the link is, the better the load balance effect is, and the whole network can also carry more traffic, that is, the bandwidth utilization rate of the whole network is high.

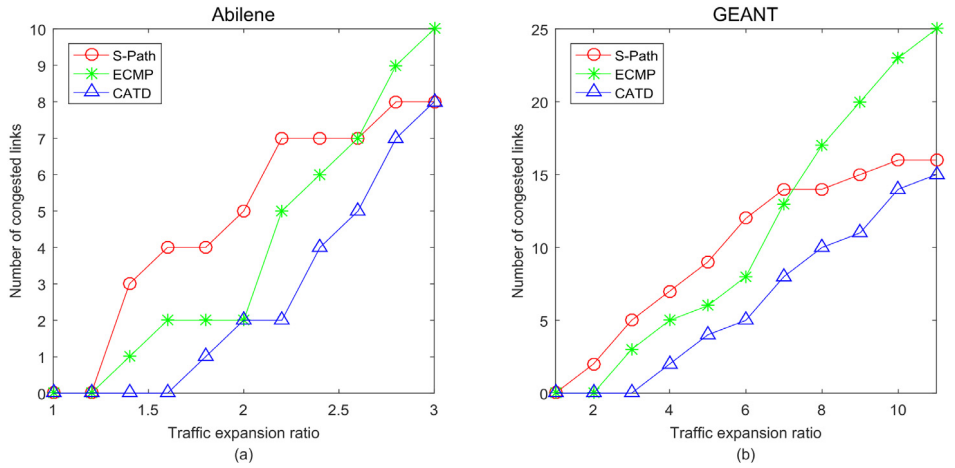


**Fig. 10.** Implementation effect comparison.

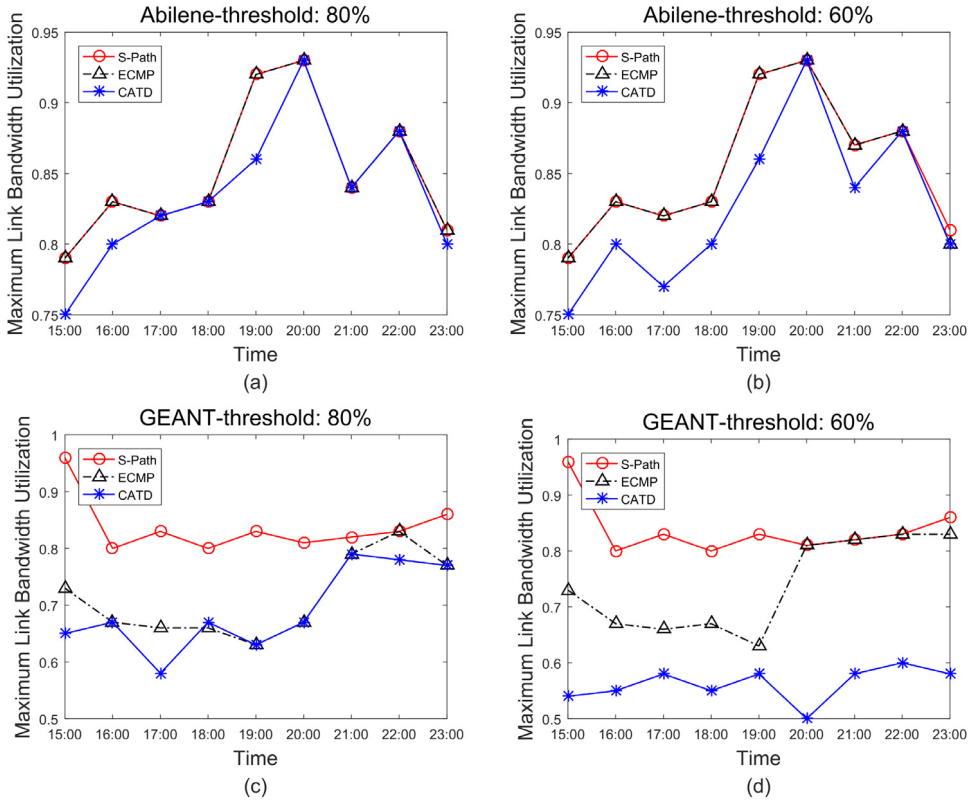


**Fig. 11.** Maximum bandwidth utilization comparison.

In the experiment, we tested the bandwidth utilization of the highest loaded link in the current network topology every hour at a certain time point. The figures (a) and (b) in Fig. 11 represent the maximum link bandwidth utilization in the Abilene topology when the dangerous thresholds are 80% and 60% respectively, and the figures (c) and (d) represent the maximum link bandwidth utilization in the GEANT topology when the dangerous thresholds are 80% and 60% respectively. We can see that, in any case, the most heavily loaded links in the CATD protected network are always the lowest. In figures (a) and (b), the result of S-Path is not obviously different with ECMP, it is because the scale of Abilene is small, which has only 12 nodes and 18 links, and the room for ECMP optimization is not that big. In figures (a) and (b), the effect of CATD is not obviously superior when the value of the other two mechanisms reach the peak. This is because the scale of Abilene topology is not large enough. By comparing with (c) and (d) of GEANT topology which has 44 nodes and 71 links, we conclude that CATD deployment has better effect on network performance and network bandwidth utilization of whole network under larger network topology. Therefore, the bandwidth of the whole network can be better used and allocated. This experiment proves that CATD can improve the bandwidth utilization of the whole network.

Results above prove that CATD can realize congestion avoidance, increasing the transmission capacity of network bandwidth.

## 6. Conclusion

Multipath transmission is the main strategy to solve the congestion problem. This paper proposes Congestion Avoidance transmission mechanism based on Two-Dimensional forwarding (CATD), implements bypassing transmission for links which are going to congest. CATD calculates and optimizes CA-Path, which minimizes the influence on other nodes and reduces the storage of two-dimensional forwarding entries. CA-Path is considered to be superior compared with Tunnels in the transmission hop. The deployment of two-dimensional forwarding minimizes bypassing granularity, properly satisfies the demand of traffic reduction, without influencing other flow which is not selected to bypass the link. The experiment shows that the CATD achieves a better performance than the conventional single-path transmission, resulting in less transmission delay at increasing bandwidth. CATD is implemented for unexpected sudden traffic congestion in the network, and long-term congestion must be resolved by bandwidth enhancement. Therefore, when the congestion is relieved, the bypassing two-dimensional forwarding traffic should return to the original forwarding path, that is, the two-dimensional forwarding path should be revoked.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

We gratefully acknowledge the support from the National Key Research and Development Program of China (2018YFB1800403) and National Natural Science Foundation of China (61872252).

## References

[1] T. Benson, A. Akella, D.A. Maltz, Network traffic characteristics of data centers in the wild, in: ACM SIGCOMM Conference on Internet Measurement 2010, Melbourne, Australia - November, 2010, pp. 267–280.
[2] K. Xu, M. Shen, H. Liu, J. Liu, F. Li, T. Li, Achieving optimal traffic engineering using a generalized routing framework, IEEE Trans. Parallel Distrib. Syst. 27 (1) (2015) 51–65.
[3] D. Kreutz, F.M.V. Ramos, P.E. Verssimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76.
[4] B.A.A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1617–1634.
[5] A. Atlas, A.D. Zinin, Basic specification for ip fast reroute: Loop-free alternates, RFC 5286, 2008, http://dx.doi.org/10.17487/RFC5286.
[6] G. Rtvri, L. Csikor, J. Tapolcai, G. Enyedi, Optimizing igp link costs for improving ip-level resilience, in: Design of Reliable Communication Networks, 2011, pp. 62–69.
[7] G. Retvari, J. Tapolcai, G. Enyedi, A. Csaszar, Ip fast reroute: Loop free alternates revisited, in: Proceedings - IEEE INFOCOM, vol. 2, no. 3, 2015, pp. 2948–2956.
[8] S. Lee, Y. Yu, S. Nelakuditi, Z.L. Zhang, C.N. Chuah, Proactive vs reactive approaches to failure resilient routing, in: Proc. IEEE Infocom, vol. 2004, no. 1, 2004, pp. 186.
[9] Z. Zhong, S. Nelakuditi, Y. Yu, S. Lee, Failure inferencing based fast rerouting for handling transient link and node failures, in: INFOCOM 2005. Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4, 2005, pp. 2859–2863.
[10] S. Bryant, C. Filsfils, S. Previdi, M. Shand, Ip Fast Reroute Using Tunnels, Tech. Rep., Internet Engineering Task Force, 2007, draft-bryant-ipfrr-tunnels-03.
[11] L. Csikor, G. Rtvri, Ip fast reroute with remote loop-free alternates: The unit link cost case, in: International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, 2013, pp. 663–669.
[12] Y. Yang, M. Xu, Q. Li, A lightweight ip fast reroute algorithm with tunneling, in: IEEE International Conference on Communications, 2010, pp. 1–5.
[13] M. Xu, L. Pan, S. Yang, Q. Li, Ip Fast Reroute Using Tunnel-at, Internet-Draft, Internet Engineering Task Force, 2010, draft-xu-ipfrr-tunnelat-01.
[14] L. Pan, M. Xu, Q. Li, J. Dan, Lightweight ip fast reroute with tunnel-at. in: International Workshop on Quality of Service, IWQOS 2010, Beijing, China, 16–18 June, 2010, pp. 1–2.
[15] M. Xu, S. Yang, D. Wang, J. Wu, Two dimensional-ip routing, in: International Conference on Computing, Networking and Communications, 2013, pp. 835–839.
[16] M. Xu, S. Yang, D. Wang, J. Wu, Efficient two dimensional-ip routing: An incremental deployment design, Comput. Netw. 59 (3) (2014) 227–243.
[17] H. Ce, Rfc 2992 analysis of an equal-cost multipath algorithm, RFC2992.
[18] N. Taft-Plotkin, B. Bellur, R. Ogier, Quality-of-service routing using maximally disjoint paths, in: Seventh International Workshop on Quality of Service, 1999, pp. . 119–128.
[19] S. Nelakuditi, Z. Zhang, D. Du, On selection of candidate paths for proportional routing, Comput. Netw. 44 (1) (2004) 79–102.
[20] A. Gopalan, S. Ramasubramanian, Ip fast rerouting and disjoint multipath routing with three edge-independent spanning trees, IEEE/ACM Trans. Netw. 24 (3) (2016) 1336–1349.
[21] H. Su, A local fast-reroute mechanism for single node or link protection in hop-by-hop routed networks, Comput. Commun. 35 (8) (2012) 970–979.
[22] K. Xu, H. Liu, J. Liu, J. Zhang, Lbmp: A logarithm-barrier-based multipath protocol for internet traffic management, IEEE Trans. Parallel Distrib. Syst. 22 (3) (2011) 476–488.
[23] J. Moy, Rfc 2328 ospf version 2, RFC2328, 28, 1998, 1.
[24] Abilene, Internet2 network advanced layer 3 service, 2017, https://www.internet2edu/media/medialibrary/2017/05/17/I2-Network-Infrastructure-Topology-Layer_3logos-201705_TnrVotx.pdf.
[25] Cernet, Cernet topology, 2017, http://www.cernet.com/aboutus/gyce_tpt.htm.
[26] Cernet2, Cernet2 topology, 2017, http://www.cernet.com/aboutus/internet2_tp.htm.
[27] Geant, Geant topology, 2017, https://www.geant.org/Resources/Documents/GEANT_topology_map_august2017.pdf.

**Wenlong Chen** born in 1976, Ph.D.,associate professor. His main research interests include Internet architecture, high performance router, IPv4/IPv6 transition and network protocol.

**Heyang Chen** is currently a postgraduate student in the College of Information Engineering, Capital Normal University. Her research interests include Internet architecture, high performance router and network protocol.

**Zhiliang Wang** Ph.D.,associate professor in the Institute for Network Sciences and Cyberspace at Tsinghua University. His research interests include formal methods and protocol testing, Internet architecture and protocols, network measurement.

**Chengan Zhao** was a post doctor at Beijing University of Posts and Telecommunications between 2014 and 2017. His research interests include computer network architecture and Internet routing.

**Mingwei Xu** born in 1971, Ph.D., professor. His research interests include network architecture, high-performance router architecture and protocol test.

**Ke Xu** born in 1974, Ph.D., professor. His research interests include architecture of next-generation of Internet, high performance router, P2P and overlay network, Internet of Things.

**Yingya Guo** is currently a PhD candidate at the Department of Computer Science and Technology, Tsinghua University. Her research interests include traffic engineering, routing optimization and Software Defined Networking.