STUDIA INFORMATICA 2003
Volumen 24 Number 2A (53)

Dániel ORINCSAY, Balázs Gábor JÓZSA, Dániel KÓSA, Levente TAMÁSI
Ericsson Research Hungary, Traffic Analysis and Network Performance Laboratory

RECOVERY USING PRECALCULATED PATHS BASED ROUTING IN ATM NETWORKS

Summary. The ATM technology serves as basis for many communication networks. The generally applied PNNI protocol enables to use sophisticated traffic engineering techniques. A progressive approach is such a routing strategy that applies precalculated paths in order minimize the reaction time for a new call request. As an advanced technique, more precalculated paths can be stored for each destination in every switch, which increases the possibility of the successful call setup. Generally, the well-known *K* shortest paths algorithm is used for this purpose that calculates the *K* cheapest paths based on an appropriate cost function. Although this method performs well in normal operation, it does not ensure that there is a bypass path among the stored ones when a network element fails along the currently used path. In this study we propose a new path calculation concept based on the algorithm of Edmonds and Karp, aiming at reducing the restoration time in case of failure. We investigate the performance of the novel method through simulations.

Keywords: ATM, PNNI protocol, *k* shortest paths algorithm, Edmonds and Karp algorithm.

PODNOSZENIE Z UPADKÓW W SIECIACH ATM PRZY WYKORZYSTANIU OBLICZONYCH WCZEŚNIEJ ŚCIEŻEK

Streszczenie. Technologia ATM służy jako podstawa wielu sieci komunikacyjnych. Stosowany w niej najczęściej protokół PNNI pozwala na wykorzystanie złożonych technik inżynierii ruchu. Coraz częściej stosuje się strategię wyboru drogi wykorzystującą wcześniejsze wyliczanie ścieżek, w celu zmniejszenia czasu odpowiedzi na żądanie nawiązania nowego połączenia. Rozwój techniki pozwala na zapamiętywanie w każdym węźle coraz większej liczby takich obliczonych uprzednio ścieżek, co zwiększa prawdopodobieństwo ich wykorzystania. Do ich obliczenia stosuje się dobrze znany algorytm *K* najkrótszych ścieżek, wykorzystujący założoną funkcję kosztów. Opisane podejście działa dobrze przy normalnej pracy sieci, nie pozwala jednak znaleźć nowych, uzupełniających ścieżek dla przełączenia ruchu, gdy zawodzi jeden z elementów aktualnie wykorzystywanej ścieżki. Niniejszy artykuł proponuje nową metodę obliczania ścieżek, opartą na algorytmie Edmondsa i Karpa, pozwalającą na zmniejszenie czasu nawiązania nowego połączenia po upadku pracującej ścieżki. Przydatność metody jest badana za pomocą modeli symulacyjnych.

Słowa kluczowe: ATM, protokół PNNI, algorytm *k* najkrótszych ścieżek, algorytm Edmondsa i Karpa.

1. Introduction

The Private Network-Network Interface (PNNI) [1] used in Asynchronous Transfer Mode (ATM) networks enables to apply dynamic and adaptive routing strategies (e.g., [2, 3]). Since PNNI does not specify the particular routing method, various approaches – including the route calculation and number of alternative routes as main issues – can be followed. Using such routing methods that can quickly adapt to significant changes in network state – e.g., increased load or network element failure – can result in high availability and low reaction time, consequently an improved network performance. In this way, the high quality requirements of customers specified by traffic contracts can be fulfilled. Generally, the sophisticated routing methods are based on the topology and performance information flooded through the network. A progressive concept is to store more precalculated paths for each destination in switches (e.g., [4]). The goal of this approach is to increase the possibility of the successful call setup, mainly in case of high network load. In the simplest case the well-known *K shortest paths algorithm* (KSP) [5] can be used to specify the paths, which calculates the *K* cheapest paths (per node-pair) based on an appropriate cost function. The classical approach is to assign a static *administrative weight* (AW) to each link. In a dynamic environment it is advisable to apply link *load based* (LB) cost functions [3] that provide good performance in normal network operation. However, the drawback of KSP is that it does not ensure a bypass path among the stored ones when a network element fails along the current path. In this study we propose a new path calculation concept whose purpose is to reduce the restoration time in case of failure without performance degradation in normal operation. The motivation for this strategy is the relative frequent occurrence of network element failures (such as link cuts) observed by the network operators [6, 7].

In the next section we present our new route calculation algorithm in details, then we investigate its performance through various simulation scenarios. Finally, we conclude our work in Section 5.

2. Routing model

2.1. Routing concept

As mentioned earlier, the base of the investigated routing environment is that switches store k precalculated paths for each destination. When a call setup request arrives the first stored path – that is the best by some consideration – is tried. If it is proved to be unsuitable due to insufficient amount of bandwidth or link cut – indicated by the *crank-back* event – the next stored route is considered and so on. The paths are recalculated whenever significant change occurs in the network announced by the so-called *flooding process*, in which the changed attributes are spread through the network. In order to prevent the network from too frequent flooding, various threshold values are set defining what means significant change. In our investigation the threshold parameters *significance level* and *minimum change* were set to 50% and 3%, respectively [1]. However, in case of dynamic network operation – mainly at high network load – the rate of flooding events can be relatively high. This might cause the frequent recalculation of paths in switches, which is basically undesirable [6, 8]. Therefore, route recalculations are scheduled in the following way. Each switch checks every time interval T whether significant change has occurred since the last check or not [8]. If there was significant change, the stored paths are recalculated; otherwise the paths are kept till the next check.

2.2. Re-routing concept

When a link fails the connections using that particular link are dropped. Since permanent connections typically correspond to leased lines and/or virtual private network (VPN) elements – whose availability is strictly specified by contracts – their immediate restorations are essential. In the current technological environment every dropped switch permanent virtual circuit (SPVC) is tried to be re-routed in the following way:

1) the SPVC is tried to be re-established using such stored paths that do not contain the failed link,
2) in case of success the re-routing process terminates,
3) the setup is retried t time after the previous trial (initially $t = T/4$),
4) t is doubled ($t := 2 \cdot t$),
5) go to Step 2.

Note that although during the first re-routing trial the stored paths may contain the failed link, later the stored paths are being continuously refreshed due to the topology information updates. Thus, it often happens that an initially blocked SPVC can be established later.

3. The new Edmonds-Karp algorithm based method (EKB)

The task of the original *Edmonds-Karp algorithm* (EK) [9] – that is based on successive executions of *Dijkstra's shortest path algorithm* [10] – is to find K (two or more) disjoint paths between a given node-pair in such way that their sum cost would be minimal according to a certain cost function. Specifying three or more disjoint paths for each node-pair in a topology of a common communication network is unreasonable and typically impossible. Moreover, the purpose of our new concept is to ensure a bypass path when a network element fails along the current path, which generally does not require fully disjoint paths. Thus, in our proposal the original EK is modified in the following way. In case of $k \geq 3$ we multiply every link of the graph by $(k-1)$. In this way $(k-1)$ parallel virtual links become available between the neighboring node-pairs, e.g., for $k = 3$ the edges of the original graph are doubled. Then EK is executed to find k disjoint paths based on this extended topology. Since $(k-1)$ parallel virtual links – representing the original one – coexist for each neighboring node-pair in the extended topology, at most $(k-1)$ of the calculated paths between a source-destination node-pair use a certain original link. Consequently, one or more paths among the k stored ones exclude the given link. In this way, when any link of the currently used path fails, it is sure that a bypass path can be found among the remaining ones. Obviously, if $k = 2$ then the original topology is considered in EK. Further, if $k = 1$ the simple Dijkstra's algorithm can be applied. Naturally, the immediate recovery is not possible in this case. We note that EK does not specify the order of calculated paths, thus in EKB they are ordered increasingly by the applied cost function.

4. Simulation

4.1. Simulation arrangement

The main focus of the current study is to examine the short-term effects of a link failure in the network, thus we applied the following simulation scenario:

1) the network is initially filled up with switched permanent virtual circuits (SPVCs) to a previously specified level of total network load,
2) a previously specified link is cut,
3) the dropped SPVCs are tried to be re-routed (in the way detailed in Section 2.2),
4) the simulation is finished if all dropped SPVCs are re-established or there is no way to re-establish any of the remained ones.

In order to get accurate results numerous network situations were tested. The network topologies were generated randomly. The network size was varied from 10 to 30 nodes, however, we detail the results of the 20-node scenarios. The average nodal degree was set to 4. The links were generally type STM-4 (622 Mbps), having administrative weight of 5040 [1]. SPVC requests were generated randomly with uniform distribution regarding the source-destination node-pairs. Their bandwidth requirements were uniformly 2 Mbps. Since SPVCs are typically long-lived, their holding time values were considered as infinite. The initial average total network load was in the interval from 40% to 70%. In order to achieve such network load values the *offered network traffic* – which is the sum of required bandwidth by connection requests – was shifted from 4 up to 7 Gbps by 0.5 Gbps steps (in the 20-node scenario). The number of stored paths per destination indicated by parameter k was varied from 1 to 5. However, we only present results for $k = 1$, 3, and 5, which well represents the tendency observed when increasing the number of stored paths. Both the AW and LB cost function approaches were considered during simulations. In case of LB the cost of a particular link e was calculated in the following way:

$$\frac{1}{2 \cdot MaxAW - AW(e)} + \max\begin{Bmatrix} load(e) - 0.9 \\ 0 \end{Bmatrix},$$

where $MaxAW$ means the largest AW value in the network, $AW(e)$ is the AW assigned to link e, and $load(e)$ is the relative load of link e scaled to interval $\{0, 1\}$. This is a modified version of the linear approximation of *implied cost-based* link cost function [3] completed with the ability of taking also the AW values into account. To ensure the reliability of results 5 different topologies, 3 different initial fill-ups per topology, and 5 different link cuts per initial fill-ups – resulting in 75 different problem instances – were investigated at each offered network traffic level. We identified two measures that well characterize the efficiency of algorithms in case of link faliure: (1) re-routing probability of dropped connections, and (2) average re-routing time of dropped connections.

4.2. Initial fill-up

In the first step of the applied simulation arrangement an initial fill-up of the network is performed. SPVCs established in this phase represent the long-life connections being permanently present and reserving relevant ratio of the whole network capacity. Considering the calculation of paths in the initial fill-up phase two approaches exist: (1) method dependent, and (2) method independent. The first approach means that during the fill-up phase the paths are calculated based on the actually examined method – including the route calculation concept (EKB/KSP), the cost concept (AW/LB), and the current set of parameter k – which enables the comparison of performances of routing algorithms in normal operation. On the other hand, while following the second approach the paths are calculated by a common method – i.e., Dijkstra's algorithm ($k = 1$) using LB cost function – thus identical initial states can be provided for the investigation of effect of a possible link cut. The purpose of this first scenario is to show that the new proposal EKB can provide as good network utilization as KSP in normal circumstances, therefore the method dependent fill-up is considered here, while method independent fill-up is used in all latter examinations enabling a more accurate comparison of the recovery capabilities of the various methods.

As we can see in Fig. 1, the new proposal EKB provides the same network utilization as KSP using cost function LB, and a bit higher utilization with AW. Further, it is clearly visible that the application of load based cost function improves the performance of routing, independently of the particular route calculation concept. Another important observation is that storing 5 paths per destination instead of 3 paths does not cause significant increase in network utilization.

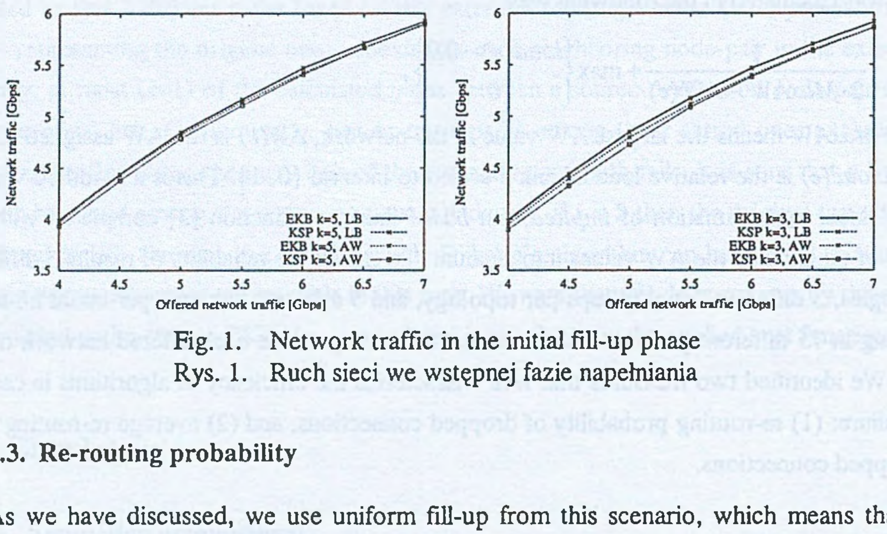
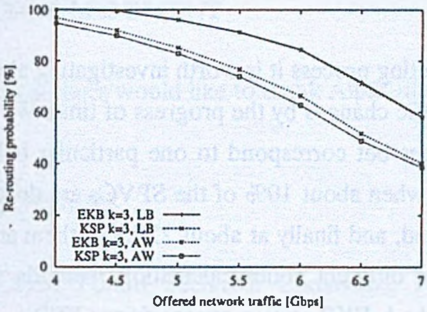


Fig. 1. Network traffic in the initial fill-up phase
Rys. 1. Ruch sieci we wstępnej fazie napełniania

4.3. Re-routing probability

As we have discussed, we use uniform fill-up from this scenario, which means that the network states are the same at the moment of link cut in all cases (at a given level of offered network traffic). Analyzing the measured re-routing probability values depicted in Fig. 2, we find that the tendency is very similar to the result of previous simulation. EKB can re-route practically the same number of SPVCs as KSP when applying LB cost function. Applying the pure AW based cost function the performance of methods is significantly lower, thus we focus only on LB cost function in the following examinations.



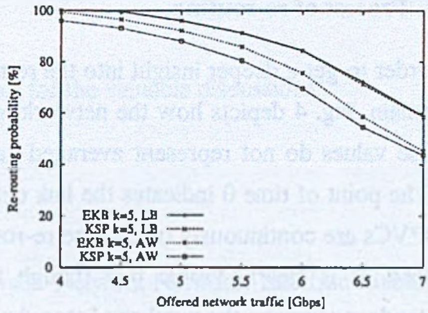


Fig. 2. Re-routing probability values
Rys. 2. Prawdopodobieństwa zmiany ścieżki

4.4. Average re-routing time

This scenario addresses the evaluation of average time needed to re-route an SPVC after link cut. The first observation is that the average re-routing time values increase as the network load increase for all methods (see Fig. 3). However, it is important to note that from 5 Gbps offered network traffic the re-routing probability is relevantly less than 100%, consequently many connections are not re-established, and in this way they are not involved in the calculation of average re-routing time. This may cause the phenomenon of reduced re-routing time at 7 Gbps offered network traffic compared to the 6.5 Gbps case. Another important tendency is that the more paths are pre-calculated and stored the less time is needed for recovery. Moreover, the improvement is more than double at some load levels. Comparing the two different route calculation concepts we may say that the new proposal EKB can provide significantly lower re-routing time values, which seems to prove the reasonableness of our approach.

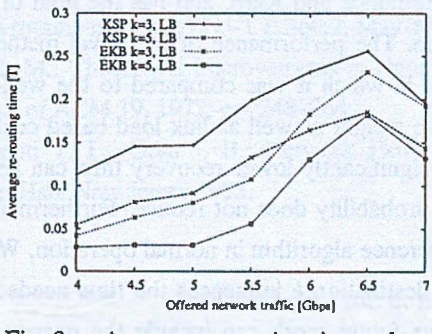


Fig. 3. Average re-routing time values
Rys. 3. Średni czas zmiany ścieżki

4.5. Process of re-routing

In order to get a deeper insight into the re-routing process it is worth investigating also the time domain. Fig. 4 depicts how the network traffic changes by the progress of time. We note that these values do not represent averaged values but correspond to one particular link cut event. The point of time 0 indicates the link cut, when about 10% of the SPVCs are dropped. Then SPVCs are continuously tried to be re-routed, and finally at about $2T$ all of them are re-established. It is clearly visible that though the different route calculation methods show similar tendency varying the number of stored paths k, EKB always overperforms KSP.

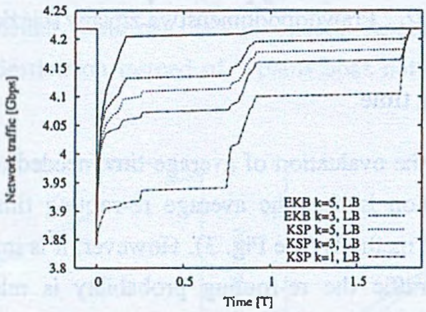


Fig. 4. Process of re-routing
Rys. 4. Proces zmiany ścieżki

5. Conclusion

In this paper we have proposed a novel route calculation method for ATM networks using PNNI protocol where precalculated paths are stored for each destination. This method is based on the algorithm of Edmonds and Karp, and has the goal of decreasing the recovery time after a single link failure. The performance of the novel method was analyzed through various simulation scenarios, in which it was compared to the well-known K shortest paths algorithm using administrative weight as well as link load based cost functions. It has turned out that using our proposal significantly lower recovery time can be achieved in case of link failure, while the re-routing probability does not reduce. Furthermore, it can provide as high network utilization as the reference algorithm in normal operation. We also examined how the number of stored paths per destination k influences the time needed to re-establish dropped connections after failure. Our future work can include the deeper analysis of the effect of different cost functions. Another interesting issue to be investigated is to test the methods on real-world network topologies.

ACKNOWLEDGMENT

The authors would like to thank Alpár Jüttner for the valuable discussions.

REFERENCES

1. ATM Forum Technical Committee: Private Network-Network Interface Specification Version 1.0. ATM Forum af-pnni-0055.000, March 1996.
2. Józsa B. G., Orincsay D., Magi Á., Szviatovszki B.: On the Use of Trunk Reservation in PNNI Routing. ICATM2000, Heidelberg, Germany, June 2000, pp. 135–139.
3. Magi Á., Szentesi Á., Szviatovszki B.: Analysis of Link Cost Functions for PNNI Routing. the 6th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, Ilkley 1998.
4. Topkis D. M.: A k Shortest Path Algorithm for Adaptive Routing in Communication Networks. IEEE Transactions on Communications, vol. 36, no. 7, July 1998.
5. Yen J. Y.: Finding the K Shortest Loopless Paths in a Network. Management Science 17, 1971, pp. 712–716.
6. Iannaccone G., Chuah C.-N., Mortier R., Bhattacharyya S., Diot C.: Analysis of link failures over an IP backbone. ACM SIGCOMM Internet Measurement Workshop. Marseilles, France, November 2002.
7. Nucci A., Schroeder B., Bhattacharyya S., Taft N., Diot C.: IS-IS Link Weight Assignment for Transient Link Failures. Sprint ATL Technical Report Nr. TR02-ATL-071000, Sprint Labs., July 2002.
8. International Telecommunication Union – Telecommunication Standardization Sector: Call routing & connection routing methods, ITU-T E.360.2, May 2002.
9. Edmonds J., Karp R. M.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of ACM 19, 1972, pp. 248–264.
10. Ahuja R. K., Magnanti T. L., Orlin J. B.: Network Flows – Theory, Algorithms, and Applications. Prentice-Hall, New Jersey 1993.

Recenzent: Prof. dr hab. inż. Tadeusz Czachórski

Wpłynęło do Redakcji 27 marca 2003 r.

Streszczenie

Technologia ATM służy jako podstawa wielu sieci komunikacyjnych. Stosowany w niej najczęściej protokół PNNI pozwala na wykorzystanie złożonych technik inżynierii ruchu. Coraz częściej stosuje się strategię wyboru drogi wykorzystującą wcześniejsze wyliczanie ścieżek, w celu zmniejszenia czasu odpowiedzi na żądanie nawiązania nowego połączenia. Rozwój techniki pozwala na zapamiętywanie w każdym węźle coraz większej liczby takich obliczonych uprzednio ścieżek, co zwiększa prawdopodobieństwo ich wykorzystania. Do ich obliczenia stosuje się dobrze znany algorytm K najkrótszych ścieżek, wykorzystujący założoną funkcję kosztów. Opisane podejście działa dobrze przy normalnej pracy sieci, nie pozwala jednak znaleźć nowych, uzupełniających ścieżek dla przełączenia ruchu, gdy zawodzi jeden z elementów aktualnie wykorzystywanej ścieżki. Niniejszy artykuł proponuje nową metodę obliczania ścieżek, opartą na algorytmie Edmondsa i Karpa, pozwalającą na zmniejszenie czasu nawiązania nowego połączenia po upadku pracującej ścieżki. Przydatność metody jest badana za pomocą modeli symulacyjnych.

Adresy

Dániel ORINCSAY: Ericsson Research Hungary, Traffic Analysis and Network Performance Laboratory, P.O.Box 107, H-1300 Budapest, Hungary, Daniel.Orincsay@eth.ericsson.se .

Balázs Gábor JÓZSA: Ericsson Research Hungary, Traffic Analysis and Network Performance Laboratory, P.O.Box 107, H-1300 Budapest, Hungary, Balazs.Jozsa@eth.ericsson.se .

Dániel KÓSA: Ericsson Research Hungary, Traffic Analysis and Network Performance Laboratory, P.O.Box 107, H-1300 Budapest, Hungary, Daniel.Kosa@eth.ericsson.se

Levente TAMÁSI: Ericsson Research Hungary, Traffic Analysis and Network Performance Laboratory, P.O.Box 107, H-1300 Budapest, Hungary, Levente.Tamasi@eth.ericsson.se