Global Journal of Computer Science and Technology
Network, Web & Security
Volume 12 Issue 11 Version 1.0 June 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Study of Routing Protocols in Telecommunication Networks

By Pawandeep Chahal & Dr. Jatinder Singh
Desh Bhagat Institute of Engg. & Mgt, Moga

Abstract - In this paper we have discussed the problem of routing in telecommunication networks and the salient characteristics of some of the most popular routing schemes. In particular, we have discussed the characteristics of adaptive and multipath routing solutions versus static and single-path strategies.

GJCST-E Classification: C.2.2

STUDY OF ROUTING PROTOCOLS IN TELECOMMUNICATION NETWORKS

Strictly as per the compliance and regulations of:

Study of Routing Protocols in Telecommunication Networks

Pawandeep Chahal [α] & Dr. Jatinder Singh [σ]

In this paper we have discussed the problem of routing in telecommunication networks and the salient characteristics of some of the most popular routing schemes. In particular, we have discussed the characteristics of adaptive and multipath routing solutions versus static and single-path strategies.

I. Routing: Definition and Characteristics

Routing can be characterized in the following general way. Let the network be represented in terms of a directed weighted graph G = (V, E), where each node in the set V represents a processing and forwarding unit and each edge in E is a transmission system with some capacity/bandwidth and propagation characteristics. Data traffic originates from one node and can be directed to another node (unicast traffic), to a set of other nodes (multicast traffic) and/or to all the other nodes (broadcast traffic). The node from where the traffic flow originates is also called source, or starting end-point, while the nodes to which traffic is directed are the final end-points, or destinations. The nodes in-between that forward traffic from sources to destinations are called intermediate, or relay, nodes. A flow is a vector in $R^{|E|}$ that for a traffic pair (s, D), $s \in V$, $D \subseteq V$, assigns a way of forwarding the data traffic from s to the nodes in D across the network while respecting the edge capacities and such that the sum of entering flows minus exiting flows at each node is null.

The general routing problem is the problem of defining path flows to forward incoming data traffic such that the overall network performance is maximized. At each node data is forwarded according to a decision policy parameterized by a local data structure called routing table. In this sense, a routing system can be properly seen as a distributed decision system.

According to the different characteristics of the processing and transmission components, as well as of traffic pattern and type of performance expected to be delivered, a variety of different classes of specific routing problems of practical and theoretical interest can be defined. For example, routing telephone calls in a network of mobile devices is a problem presenting characteristics which are quite different from those of the problem of routing telephone calls in a cable telephone network, which, in turn, is a problem much different from the problem of routing data packets in a best-effort connectionless data network as the Internet.

An important difference between routing and the combinatorial problems that have been considered so far consists in the presence of input data traffic which characterizes the problem instance. That is, the routing problem is composed of two parts: (i) the communication structure, which in a sense defines the constraints, and (ii) the traffic patterns that make use of this structure. It is always necessary to reason taking into account the two aspects together. For instance, the set of all the disjoint shortest paths (taken with respect to link bandwidths and propagation times) between all the network node pairs is not, in general, the optimal solution to the routing problem at hand. The optimal solution is obtained by considering the specific temporal and spatial distribution of the input traffic taken as a whole and solving simultaneously all the shortest path problems related to all the source–destination pairs relevant for traffic data. In fact, each allocated path flow recursively interferes with all the other path flows since it reduces the capacity which is available along the used links. Therefore, in a sense, the order of path flows allocation does really matter, as well as the possibility of rerouting path flows over time. That is, the knowledge about the characteristics of the input traffic is a key aspect to allow optimizing the allocation of the path flows in order to obtain optimal network-wide performance. On the other hand, in the case of routing this is rarely the case, since the characteristics of the incoming data traffic are hardly known with precision in advance. In the most fortunate cases, only some statistical knowledge can be assumed.

In practice, the routing problem in telecommunication networks must be solved online and under dynamically changing traffic patterns whose characteristics are usually not known in advance and recursively interact with the routing decisions. Moreover, routing is a fully distributed problem, a characteristic that usually rules out the use of global knowledge and/or centralized actions, and introduces problems of perceptual aliasing [15] (or hidden networks state) from the point of view of the nodes. Performance metrics usually consists of multiple conflicting objectives constrained by the specific characteristic of the

Author α : Research Scholar, CMJ University, Shillong , Meghalaya-793 003, Asst. Prof, Department of IT, Desh Bhagat Institute of Engg. & Management, Moga.
Author σ : Prof. , Dean, Desh Bhagat Institute of Engg. & Management, Moga.

transmission and processing technology. Finally, routing is a business-critical activity, therefore, any implementation of a routing system is required to be efficient, fault-tolerant, reliable, secure, etc.

In figure.1 traffic data must be forwarded from the source node 1 to the target node 13. Several possible paths are possible. Each node will decide where to forward the data according to the contents of its routing table. One (long) path among the several possible ones is showed by the arrows.

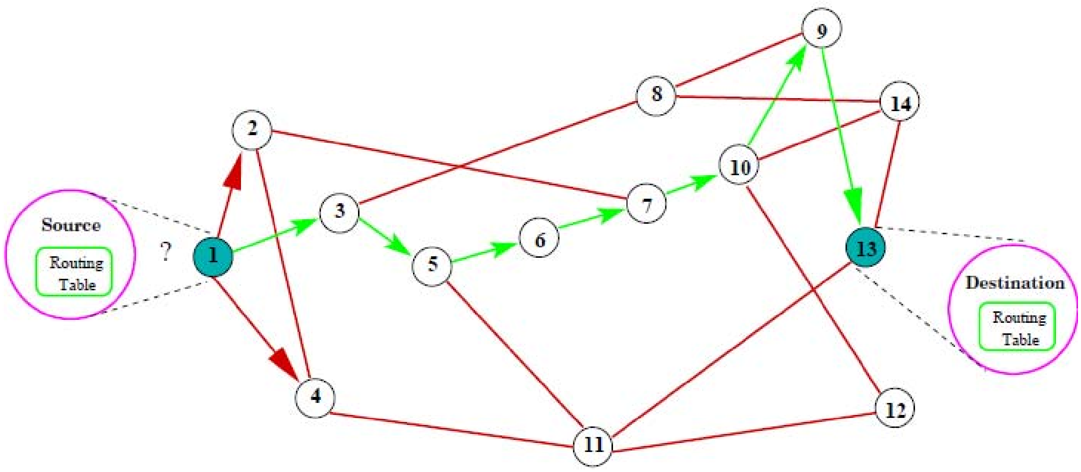

Figure 1 : Routing in networks

It is apparent that these characteristics do not find any counterpart in the class of static combinatorial problems considered so far. To have an idea, a VRP that could share a similar level of complexity, should have an unknown distribution of customer arrivals, a tight interaction among the vehicles (sort of traffic jams), strict time windows, backhauls, and the possibility for the drivers to get only local information.

When the characteristics of the traffic flows are known in advance, the problem can be solved in a centralized way, and other additional simplifications are possible, routing can be framed in the general terms of a multi-commodity flow problem, which is an important class of problems modeling the transfer of commodities from source locations to destinations.

At each network node:

- Acquisition and organization of up-to-date information concerning the local state, that is, information on the local traffic flows and on the status of the locally available resources.
- Build up a view of the global network state, possibly by some form of exchanging of the local state information.
- Use of the global view to set up the values of the local routing table and, consequently, to define the local routing policy with the perspective of optimizing some measure of network performance.
- Forward of the user traffic according to the defined routing policy.
- Asynchronously and concurrently with the other nodes repeat the previous activities over time.

II. Routing Algorithms Classification

Routing algorithms are usually designed in relationship to the type of both the network and the services delivered by the network. Under this perspective, given the variety of possible network types and delivered services, it is hard to identify meaningful and exhaustive classifications for routing algorithms. Therefore, the algorithms are classified according to few very general characteristics that can be singled out. Additional general characteristics that can be used to classify routing algorithms can be derived by the meta-algorithm, which suggests that different choices for either the optimization criteria or the strategies for building and using the local and the global views can result in different classes of algorithms. In particular, since the strategies for building the local and global views are strictly related to the way both traffic information and topological information are managed in order to define the routing tables, a classification of the different routing systems is precisely given according to the algorithm behavior, which can be static or adaptive with respect to topology and/or traffic patterns. Moreover, since different choices in the criterion to be optimized can generate different classes of algorithms, a further classification is given in this sense, making a distinction between optimal and shortest path routing. A final classification is drawn according to the number of paths that are used or maintained for the same traffic session or destination. In this sense, algorithms are divided in single-path, multi-path and alternate-path.

a) Architecture: Centralized vs. Distributed

In centralized algorithms, a main controller is responsible for the updating of all the node routing tables and/or for every routing decision. Centralized algorithms can be used only in particular cases and for small networks. In general, the controller has to gather information about the global network status and has to transmit all the decisions/updates. The relatively long time delays necessarily involved with such activities, as well as the lack of fault-tolerance (if not at the expenses of redundant duplications), make centralized approaches unfeasible in practice. From now on only non-centralized, that is, distributed routing systems are considered. In distributed routing systems, every node autonomously decides about local data forwarding. At each node a local routing table is maintained in order to implement the local routing policy. The distributed paradigm is currently used in the majority of network systems.

b) Types of routing tables: Static vs. Dynamic

Routing tables can be statically assigned or dynamically built and updated. It is evident that the performance of the two approaches can be radically different, and the appropriateness of one approach over the other tightly depends on the characteristics of the network scenario under consideration.

In both cases routing tables are built in order to possibly optimize some network-wide criteria which are made depending in turn on costs associated to network elements. That is, to each link, or whatever network resource of interest (e.g., available processing power of a routing node), a value (integer, real, nominal, etc.), here called cost, is assigned according to some metric in order to have a measure of either utilization level or physical characteristics (e.g., bandwidth, propagation delay). Therefore, the process of finding routing paths optimized with respect to the chosen criteria can be actually intended as the minimization process with respect to the defined costs (e.g., the overall cost criterion can be expressed in terms of a sum of the link costs or of the path/link flows). If trusting information about the incoming traffic patterns is available, then an optimal routing approach (i.e., a multi-commodity flow formulation) can be used to actually carry the minimization, otherwise other approaches, like those based on independent shortest path calculations, are called for.

Static routing: In static (or oblivious) routing systems, the path to forward traffic between pairs of nodes is determined without regard to the current network state. The paths are usually chosen as the result of the offline optimization of some selected cost criterion. Once defined the paths to be used for each source-destination pair, data are always forwarded along these paths.

Costs and accordingly, routing tables, are assigned either by an operator or through automatic procedures independently from the current traffic events. The use of the links' physical characteristics is one of the simplest ways to assign static link costs (e.g., a link with characteristics of high bandwidth and low propagation delay will have associated a low cost). For instance, the cost default value of a link for the Internet intra-domain protocol Open Shortest Path First (OSPF) [55, 54] as automatically assigned by most CISCO routers is 108 /b, with b being the unload bandwidth of the link [56].

Routing tables can be also assigned on the basis of some a priori knowledge about the expected input traffic. For instance, traffic statistics can be periodically recorded, and if some regularity can be spot, these can be used in turn to model the incoming traffic and assign the routing tables as the result of optimal routing calculations.

Dynam ic routing: Dynamic (or adaptive) routing goes beyond static routing by admitting the possibility of building/changing the routing tables online according to the current traffic events. It is useful to distinguish between the ability of adapting to the changing traffic conditions and to topological modifications (e.g., link/node failures, link/node addition/removal).

Topological adaptivity is in a sense more fundamental. It is not reasonable to think that every resource addition/removal should be explicitly notified by the human operator. Instead, is a minimal requirement to ask the distributed routing system to have the ability to automatically get aware of such modifications? This is what actually happens in most of the currently used routing protocols. Clearly, different protocols react in different way to such events. For instance, Bellman-Ford algorithms, since they do not make explicit use of global network topology and only use the notion of distance, suffer the problem of the so-called counting-to-infinity, that is, when a link becomes suddenly unavailable, in the worst case it might take infinite time to adjust the routing tables accordingly.

On the other hand, the most common intra-domain routing protocol, OSPF [55], is a shortest path algorithm based on topology broadcast and is able to be fully and efficiently adaptive with respect to topological modifications. However, OSPF is not really adaptive with respect to traffic modifications, such that link costs are static, and may change only when network components become unreachable or new ones come up.

As another example, the Enhanced Interior Gateway Routing Protocol (EIGRP), which is the CISCO's proprietary intra-domain protocol, is an extension of the Bellman-Ford based on the DUAL algorithm [34], such that it overcomes the counting-to-infinity problem and uses link costs which are dynamically assigned according to the following formula:

$$C = \left[k_1 B + \frac{k_2}{256-L} + k_3\right] \frac{k_5}{R-k_4} \quad (1)$$

Where k_i, i = 1, . . . , 5 are constants, L is the link load assigned as an integer over a scale going from 1 to 255. D is the topological delay, that is, the amount of time it takes to get to the destination using that link in case of unloaded network. R is the reliability of the path expressed as the fraction of packets that will arrive at destination undamaged and $B = 10^7 / \min_i b_i$, where b_i is the bandwidth of path to destination. The parameters B and D are defined during the router configuration, while L and R are estimated through measurements. However, the default link cost is also defined as C = B + D.

Generally speaking, adaptivity to traffic events is commonly obtained by monitoring local resource utilization (usually in terms of link costs), building up statically estimates of these costs, using these costs to update the local routing table and possibly exchanging this information with other nodes in order to allow some form of dissemination of fresh local information. The nature of the local statistical information and the modalities of information exchange characterize the different algorithms.

Adaptive routers are, in principle, the most attractive ones, because they can adapt the routing policy to varying traffic conditions. As a drawback, they can cause oscillations and inconsistencies in the selected paths, and, in turn, these can cause, circular paths, as well as large fluctuations in measured performance. Stability and inconsistency problems are more evident for connection-less than for connection-oriented networks. The problems with adaptive routing are well captured by the following sentence, slightly changed from the original citation: Link arrival rates depend on routing, which in turn depends on arrival rates via routing selected paths, with a feedback effect resulting.

Intuitively, the general non stationarity of the traffic patterns, as well as the above feedback effect, generate non-trivial problems of parameters setting in any adaptive algorithm. If the link costs are adaptively assigned in function of the locally observed traffic flows, which is, for instance, the amount of the variation in the traffic flows that should trigger an update of the link costs and in turn of the routing table. Should every update trigger a transmission of the new costs/routing table to other nodes in the network. In general, every answer to these questions will contain some level of arbitrariness. In fact, the values assigned to the parameters of the algorithm define the tradeoff between reactivity to local traffic changes and stability in the overall network response.

c) Optimization criteria: Optimal vs. Shortest paths

Shortest path routing is the routing paradigm most in use in real networks. In shortest path routing the optimizing strategy for path flows consists in using the minimum cost paths connecting all the node pairs in the network, where the paths are calculated independently for each pair. That is, shortest path routing adopts a per pair perspective. On the other hand, optimal routing, which is the other main reference paradigm (at least from a theoretical point of view), has a network-wide perspective, since the path flows are calculated considering all the incoming traffic sessions. Clearly, in order to adopt such a global strategy, optimal routing requires the prior knowledge of the statistical characteristics of all the incoming flows, a requirement which is usually quite hard to satisfy.

According to an optimization perspective, a more coarse-grained distinction can be also made between minimal and non-minimal routing algorithms. Minimal routers allow packets to choose only paths which are minimal with respect to some cost criterion, while in non-minimal algorithms packets can be forwarded along any of the available paths according to some heuristic decision strategy [9]. Both optimal and pure shortest path routing implement minimal routers. On the other hand, ACO algorithms for routing are not minimal, due to the presence of stochastic components playing a major role in decision-taking.

d) Load distribution

Data traffic toward the same destination d can be forwarded along always the same link or it can be spread along multiple paths. Actually, when routing tables are updated being adaptive to traffic patterns, the resulting effect can be that of actually spreading the data packets toward the same destination over multiple paths at the same time, if the updating interval is shorter than or comparable to the inter-arrival time of the packets directed to d. However, this is a quite particular and unlikely case, while, more precisely:

Multipath and alternate path routing: With multipath routing is intended the situation in which multiple next hop entries for the same destination are maintained in the routing table and used to forward data according to some (usually distance-proportional) scheme.

On the other hand, alternate routing is the situation in which information about multiple paths is maintained in the routing table but is used only as a backup in the case the primary path becomes unavailable because of failure or suddenly congested such that its quality scores poorly.

Multipath routing can be effectively visualized in the terms of defining through the distributed routing tables, instead of a collection of single paths between

Global Journal of Computer Science and Technology (E) Volume XII Issue XI Version I June 2012

each source and destination, a directed, possibly acyclic, graph rooted at the destination. Figure 5.2 graphically shows the situation. The directed links represent the available routing alternatives for packets bound for d according to the local routing tables. The leftmost graph shows a global distributed assignment of the routing tables that results in multiple loop-free paths connecting each source s_i, i = 1, 2, 3 to the destination d. The rightmost graph shows the routing table assignment that would result from a single-path shortest path routing algorithm. It is evident the difference in resources utilization in the two cases. With the single-path policy only three links are actually going to be used to forward packets toward d. This means that if the traffic rate at one of the three sources is higher than the bandwidth of the single link, either packet must be dropped or they will incur high delays. In the multipath case, the effective bandwidth available to each source is much higher, and the whole network bandwidth can be fully exploited through statistical multiplexing of link access. Clearly, in the case of lightly loaded network, when for instance the bandwidth of each single link is able to carry to whole traffic of each source, the single-path assignments will provide the best performance in terms of both maximal throughput and minimal end-to-end delays.

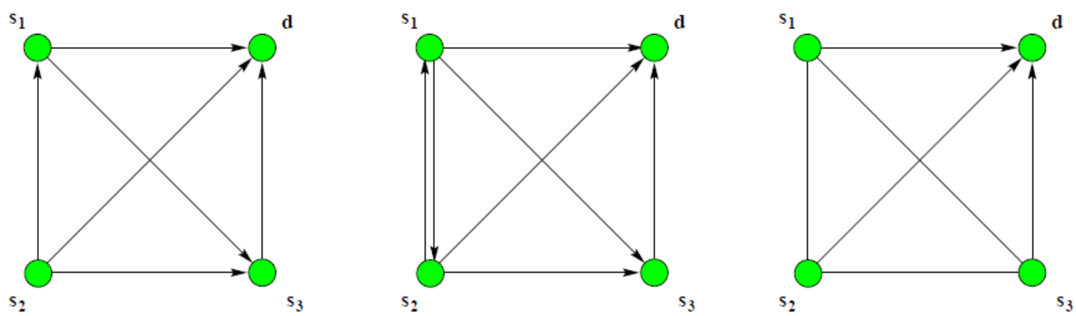

Figure 2 : Multipath routing from sources s_i, i = 1, 2, 3 to destination d

In the figure 2 the directed links show the possible routing decisions that are available at nodes for a packet bound for d according to their routing tables. The links are assumed to have all the same unit cost. The leftmost graph shows a routing policy which is globally loop-free independently from the specific policy adopted to locally spread the data along the different links. That is, the combination of the routing policies of all the nodes defines a directed acyclic graph rooted in d. The middle graph shows an assignment of the routing tables which can give rise to packet looping between $s_1$and s_2, depending on the specific utilization of the local multiple alternatives as a function, for instance, of the distance to the destination. If the distances/costs are calculated in a wrong way, possibly because of traffic fluctuations, is easy to incur in packet looping in this case. The rightmost graph shows the assignment of the routing tables resulting from a single-path shortest path calculation [8].

The multipath solution will likely show also maximal throughput, but the end-to-end delays will be worse than those of single-path since some packets will be forwarded along routes that are longer than one hop. The middle graph of the figure points out another potential drawback in multipath routing: loops can easily arise because of "wrong" composition of the local routing policies. In the case of the figure, packets can bounce between s_1 and s_2 according to the policy adopted to spread data over the available multipath and to the costs that are assigned to the different links in the perspective of reaching d.

There are the three key design issues in multipath routing protocols [7]: (i) how many paths are needed, (ii) according to which criterion these paths are selected, (iii) which data distribution policy is adopted to use the selected paths. Issues (i) and (ii) are by far the most important ones since determine the final performance of the algorithm.

Regarding (i), is clear that the optimal answer would depend on the characteristics of the both the network and traffic. However, the general target is to get good load balancing while using a low number of paths. In fact, a high number of paths bring more complexity in the management of the routing tables and increases at the same time the probability of packet looping.

The criteria to select the paths referred in point (ii) differ from network to network. Paths might be selected not only according to their quality in the sense of distance/cost to the destination, but also according to other features, like the level of node and/or edge disjointness. Disjoint paths are in principle the most appealing ones, since they allow an effective and not interfering distribution of the load. On the other hand, the need for disjointness is strictly related to the packet production rate of the traffic sources. For low rates (inferior to the links' bandwidths) it might be not really necessary to search for disjoint paths since packets for the same destination will likely not interfere along the common parts of the followed paths. On the other hand, this might be the case for destinations which are hot spots and concentrates high rates of traffic from several sources. The issue of disjointness is particularly

Global Journal of Computer Science and Technology (E) Volume XII Issue XI Version I June 2012

important in the case of connection oriented networks providing quality of service, since disjointness means also increased robustness to failures for the single session: if multiple paths toward the same destination share several networks elements, the failure of one of these elements will cause the breakdown of the whole bundle of paths and consequently of the QoS session. Disjointness is even a more critical issue in the case of mobile ad hoc networks. In fact, in presence of high rates of data generation the use of multiple paths can be effective only if the paths are radio-disjoint. If this does not happen, packets from the same session hopping between different nodes situated in the same radio range will likely generate MAC-level collisions when accessing the shared radio channel. As a result, the use of multiple paths can in principle dramatically bring down the performance, instead of boosting it. In general quite difficult to identify disjoint paths. This is true in particular for mobile ad hoc networks, because of the highly dynamic conditions, and in connection-less networks, like the IP networks, since every routing table is built according to a local view and routing decisions are taken independently at each node, while it might be quite straightforward to do in connection-oriented networks. Referring to the last considered point (iii), the policies adopted to spread data on the available paths usually follow a proportional approach based on the estimated cost/quality of the paths. That is, each link is used for a destination proportionally to the estimated quality of the associated path toward that destination. This is the approach followed for instance in the Optimized MultiPath (OMP) [9] scheme, in which link load information is gathered dynamically. On the other hand, the Equal Cost MultiPath (ECMP) strategy [5] adopted by OSPF on the Internet, consists in considering only the set of paths with equal (best) quality and distributing the traffic evenly among them. In variance-based approaches [6] if J_{min} is the cost associated to the best path among the locally known ones, then all paths whose cost is $J \leq vJ_{min}$, $v \geq 1$, are used for routing, depending on the specific value of the "variance" parameter v. In EIGRP the traffic is split over these paths proportionally to their metric.

The use of multipaths appears as particularly appealing in the case of QoS networks, since it can bring significant advantages during both the connection setup phase, when the requested resources must be found and reserved, and the data communication phase. In fact, at setup time, multiple concurrent reservation processes can be used for the same session [16], such that (a) the search can be speed up since multiple paths are tried out at the same time, (b) a failure in one or more of the processes does not affect the others, and (c) if several routes are made available for reservation the most appropriate one(s) can be selected. During the session running time, the availability of multiple paths can allow an easier recovering from link or node failures, as well as the shifting and/or splitting of the connection flow over other paths in order to gracefully adapt the load distribution and possibly minimizing the blocking probability for the forthcoming sessions. The positive features provided by the use of multipath routing at setup time suggest that it can play an important role especially to allocate bursty applications, as it is also confirmed by theoretical analysis in [7]. Interestingly, also the theoretical analysis in [6, 7], which refers to the use of multipath for best-effort routing in the IP networks, suggests that multipath can bring significant advantages to deal with bursty connection (while the long-lived connections, which account for the majority of the Internet traffic, preferentially should not be split over multiple paths).

A potential drawback of adopting a multipath strategy consists in the fact that if the data packets of the same traffic session are spread over different multiple paths, each associated to a possibly different traveling time, packets will likely arrive at destination out-of-order, creating problems to the transport protocol. For instance, facing such a situation, a TCP-like algorithm could easily get wrong and start asking for packet retransmissions while packets are just arriving out-of-order and slightly time-shifted. A solution to this problem could consist in hashing at the routing layer of each intermediate node the TCP connection identifiers (source and destination IP addresses) of each received packet in order to determine the next hop [56, 78]. In this way, packets from the same source/application are always forwarded along the same outgoing link, while the overall load is however balanced since different TCP connections are routed along possibly different links. This solution has the drawback that in case of few long-lived heavy loaded traffic sessions, network utilization can be result quite close to the single-path case, losing in this way the possibly advantages of using a multipath protocol. Moreover, if the number of traffic sessions is high, the memory requirements necessary to keep trace of all the hashed values might result unfeasible (at least for most of the current commercial routing boxes which are equipped with a limited small amount of memory). In more general terms, one might think that if multipath routing is used then the transport layer algorithms should be consequently adapted in order to fully exploit the potentialities of using multipath at the routing layer.

III. Optimal and Shortest Path Routing

Shortest path routing is the most popular form of routing strategy in current data networks. Therefore, it is customary to review in detail the characteristics of this class of algorithms. On the contrary, optimal routing algorithms are extremely important from a theoretical point of view, since they provide a solution which is globally optimal.

a) Optimal routing

Optimal routing has a network-wide perspective and its objective is to optimize a function of all individual link flows. Optimal routing models are also called flow models because they try to optimize the total mean flow on the network. They can be characterized as multi commodity flow problems, where the commodities are the traffic flows between the sources and the destinations, and the cost to be optimized is a function of the flows, subject to the constraints of flow conservation at each node and positive flow on every link. Obviously, the flow conservation constraint can be explicitly stated only if the arrival rate of the input traffic is known and if no packets can be dropped. The routing policy consists of splitting any source-target traffic pair at strategic points, then shifting traffic gradually among alternative routes. This usually results in the use of multiple paths for a same traffic flow between the same origin-destination pair and in conditions of load balancing.

The multi commodity flow model of an optimal routing problem is solved with respect to the so-called path flow variables x_p :

$$min \sum_{<i,j>} G_{ij} \left[\sum_{\substack{all\ paths\ p \\ containing <i,j>}} x_p \right]$$

$$\sum_{p \in P_w} x_p = r_w , \qquad \forall\, w \in W$$

$$x_p \geq 0 \qquad \forall\, p \in P_w ,\ w \in W \tag{2}$$

Where W is the set of all origin-destination pairs in the network, r_w is the known input traffic rate of the origin-destination pair w ∈ W, and P_w is the set of all directed paths that can connect the w's origin-destination nodes. G_{ij} is the cost function associated to the data flow on the link <i, j>. The overall function to minimize is the sum of all these G_{ij}, that is, a function of the overall cost associated to all the assigned path flows x_p. The form of G_{ij} is left uninstantiated in the formula. According to the different characteristics of the network and of the provided services, each G_{ij} can be chosen in a variety of different ways. If multiple conflicting objectives have to be taken into account, it might result quite hard to define an additive function $G = \Sigma\, G_{ij}$ which is able to capture all of the objectives. In general terms, it is preferred to choose a functional form of G such that the problem can be solved with analytical methods, usually by derivation operations. A common choice for G consists in:

$$G_{ij}\left(F_{ij}\right) = \frac{F_{ij}}{C_{ij} - F_{ij}} + d_{ij} F_{ij} , \tag{3}$$

Where the C_{ij} are related to the capacity of the link, the d_{ij} are the propagation delays, and F_{ij} is the flow through the link < i, j >. According to this formula, the cost function becomes the average number of packets in the network under the hypothesis, usually not valid in real networks that each queue behaves as an M/M/1 queue of packets. However, when formula 5.3 is used and under the M/M/1 hypothesis, the sum of the G_{ij} is the total delay experienced by data packets. Gallager proposed an algorithm to carry out these computations in a distributed way while ensuring also loop-freedom at every instant. Unfortunately, the algorithm critically depends on a global step-size parameter which depends in turn on the specific characteristics of the input traffic patterns. Such that the algorithm of Gallager can be used in practice only to provide lower bounds under stationary traffic. The cost function G can be also alternatively expressed not as a sum of functions G_{ij}, but also, for example, as a max-norm:

$$G = \max_{<i,j>} \left\{ \frac{F_{ij}}{C_{ij}} \right\} ,$$

However, in these cases it is usually more difficult to solve the problem analytically.

b) Shortest path routing

Shortest path routing has a single origin-destination perspective. The path between each node pair is considered in isolation from the paths for all the other pairs. In this sense, the shortest path perspective is opposed to that of optimal routing, which makes use of a cost function of the flows of all the origin-destination pairs considered altogether. No a priori knowledge about the traffic process is required, although such knowledge can be fruitfully used, when available.

Main characteristic of shortest path routing: In shortest path algorithms, at each node s, the local link which is on the minimum cost path to the destination d, for all the possible destinations d in the network is identified and used to forward the data traffic directed to d. The minimum cost path is calculated without taking into account the paths for the other destinations. That is, the path for each destination is treated as an entity independent from the paths (i.e., the paths flows) for all the other destinations. This is in contrast with the optimal routing approach that allocates each flow minimizing a joint function of all the flows in the network. The general common behavior of most implementations of shortest path algorithms is informally described in Algorithm 2:

At each network node:

1. Assign a cost to each one of the out links. The cost can be either static or adaptive; in the following it is assumed the most general case of adaptive link costs.
2. Periodically, and without the need for inter-node synchronization, transmit to the neighbors either estimates about cost and status (on/off) of the attached links, or some other information related to the estimated distance/delay from the node to the other known nodes in the network.
3. Upon receiving fresh information from a neighbor, update the local routing table and local information database (i.e., the local view of the global network status). The routing tables are updated in order to associate to each destination the out link that satisfies the conditions of minimum cost path. That is, for each network destination d, the out link belonging to the minimum cost path to reach d will be used to route data traffic bounded for d. The computation of the minimum cost paths is executed on the basis of the locally available information only.
4. The received information packet, and/or the updated routing information, can be in turn also forwarded to the neighbors, which might further forward it.
5. Data routing decisions are made according to a deterministic greedy policy by always choosing the link on the minimum cost path.
6. Asynchronously and concurrently with the other nodes repeat the previous activities over time.

Algorithm 2: General behavior of shortest path routing algorithms.

The general scheme of Algorithm 2 mainly addresses single-path algorithms. Multipath implementations can be realized by building and maintaining at each node information about more than one path toward each destination. Accordingly, the routing decisions at point 5 can be such that either all the equally good paths are considered for use, or also non-minimal strategies are adopted, such that a set of the n best paths are used in some way.

According to the different contents of the routing tables, shortest path algorithms can be further subdivided in two major classes termed distance-vector and link-state [2]. The following two subsections are devoted to the description of the characteristics specific to each class.

i. *Distance-vector algorithms*

In distance-vector algorithms, each node n maintains a matrix $D_d^n(i)$ of distance estimates for each possible network destination d and for each possible choice of next node i, where i ∈ N(n), the set of neighbor nodes of n. These distance estimates are used to build up the vector SD_{nd} of the shortest distances to d, which, in turn, is used to implement routing decisions. Hereafter, distance is to be intended in a general sense as an additive cost-to-go to reach the destination node. Figure 5.3 shows all the components of generic distance-vector schemes.

The stored topological information is represented by the list of the known nodes identifiers. The average memory occupation per node is of order O(Nn), where N is the number of nodes in the network and n is the average connectivity degree (i.e., the average number of neighbor nodes considered over all the nodes). Distance-vector algorithms forward a packet with destination d along the local link belonging to the path associated with the shortest estimated distance SD_{nd} to d. Therefore, the central component of the algorithm is the distributed computation of such minimum cost paths using the locally available topological description of the network, the costs-to-go received from the neighbors, and the local distance to the neighbors.

The framework of the distributed (asynchronous) dynamic programming provides and optimal and efficient way of carrying out the required computations given the topological description available at each node. The basic idea is the association of each node with a state of a DP backward algorithm. The value of each state n for each destination d, is the estimated shortest distance SD_{nd} from n to d. Link choices correspond to state actions. The resulting algorithm, the basic distributed Bellman-Ford algorithm (DBF) [4, 33], works in an iterative, asynchronous and distributed way. Every node n assigns, in a static or dynamic way, a cost to its local links.

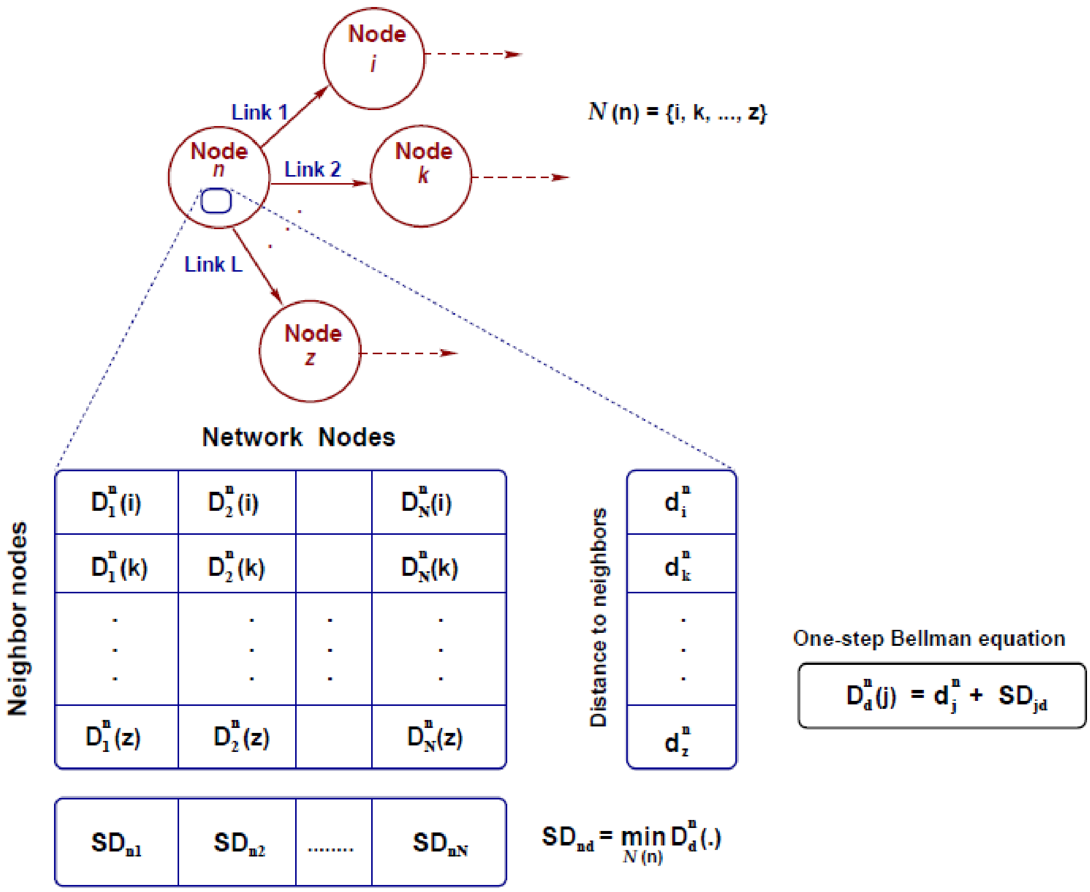

Figure 3 : Data structures and basic equations used in distance-vector algorithms

On the basis of this cost, the cost to travel (the "distance") d_i^n to each of the physically connected neighbors i ∈ N(n) is consequently defined. This one-step distance is used, in turn, within a one-step Bellman equation in order to compute/estimate the traveling distance to each one of the possible destinations d in the network for each one of the local next hops i:

$$D_d^n\ (i) = d_i^n + SD_{id} \tag{4}$$

Once the entries of the matrix D are set up, the vector SD of the shortest distances from n is set up accordingly:

$$SD_{nd} \ = \min_{j \in N(n)} \left[d_j^n + SD_{jd} \right] \tag{5}$$

The routing table is defined at the same time as the vector SD: for each destination d the chosen next hop node is the one minimizing the equation 5.5 used to compute SD.

Clearly, each node n, in order to compute the matrix of the estimates D, in addition to the locally estimated value d_i^n, needs to know the values SD_{id} from all its neighbors i ∈ N(n). This is the critical part of the distributed algorithm. At the beginning of the operations, the matrix D and the vector SD are initialized all over the network nodes with the same arbitrary values. Then, at each node n, when either the local cost estimates are updated, or an updated value of SD is received from one of the neighbors, the Equations 4 and5 are re-computed, the routing table is updated, and the possibly new value of SD_{nd} is sent, in turn, to all its neighbors. Iterating this distributed asynchronous behavior over the time, after a transitory phase, the distance estimations at each node converge to the correct minimum values with respect to the used cost metric. More precisely, the algorithm always converges and converges fast if the link costs, that is the distances d to the neighbors, are either stationary or decrease [3]. On the other hand, convergence is not anymore assured if link costs increase, or, when link failures result in network partitions the algorithm never convergence. This is the well-know problem of counting-to-infinity, which results from the fact that it might happen that using the distance communicated by a neighbor, a node computes in turn its distance to a destination on the basis of the length of the path passing through itself. Clearly, the node using this "circular" distance is unaware of the circularity since nodes only exchange distance and no path information.

IV. Conclusion

All the adaptive algorithms considered in the chapter gather traffic load information only according to a passive strategy. That is, it is common practice to monitor at the nodes the load associated to each attached link in order to update statistics that are in turn used either to compute distances or are broadcast to the other nodes. On the other hand, there is no notable example of gathering information according to also an active strategy. For example, by generating an agent and sending it into the network with the purpose of collecting some useful information about a well defined resource or destination.

Taking into account all the aspects discussed so far, it is possible to compile a sort of wish list for the design characteristics of novel routing algorithms, that are expected to: (i) be traffic adaptive, (ii) make use of multipaths, (iii) integrate both forms of collective rationality and continual and graceful adaptation of the routing policy, (iv) show robustness with respect to parameter setting, with possible self-tuning of the parameters in order to adapt to the characteristics of the specific network scenario, (v) limit loop formation, or at least ensuring that loops are very short-lived, (vi) possibly not fully rely on information bootstrapping or broadcasting, in order to obtain more robustness under dynamic and near saturation conditions, while at the same time providing at least near-optimal performance under static and low load conditions, (vii) make use of stochastic components in order to be more robust to the lack of global up-to-date information at the nodes, (viii) implement some form of (pro)active information gathering to complement passive information gathering one, while at the same time limiting the associated routing overhead. Our ACO algorithms for routing have been precisely designed according to these guidelines, resulting in novel traffic-adaptive algorithms for stochastic multipath routing.

Bibliography

1. T. Abe, S. A. Levin, and M. Higashi, editors. Biodiversity: an ecological perspective. Springer-Verlag, New York, USA, 1997.
2. M. Beckmann, C.B. McGuire, and C.B. Winstein. Studies in the Economices of Transportation. Yale University Press, 1956.
3. R. Bellman. Dynamic Programming. Princeton University Press, 1957.
4. R. Bellman. On a routing problem. Quarterly of Applied Mathematics, 16(1): 87–90, 1958.
5. D. Bertsekas, J. N. Tsitsiklis, and Cynara Wu. Rollout algorithms for combinatorial optimization. Journal of Heuristics, 3(3):245–262, 1997.
6. M. Birattari, G. Di Caro, and M. Dorigo. Towards the formal foundation of ant programming. In M. Dorigo, G. Di Caro, and M. Sampels, editors, Ants Algorithms - Proceedings of ANTS 2002, Third International Workshop on Ant Algorithms, Brussels, Belgium, September 12–14, 2002, volume 2463 of Lecture Notes in Computer Science, pages 188–201. Springer-Verlag, 2002.
7. J.A. Bland. Optimal structural design by Ant Colony Optimization. Engineering optimization, 33: 425–443, 2001.
8. J.A. Bland. Layout of facilities using an Ant System approach. Engineering optimization, 32(1), 1999. [38] J.A. Bland. Space-planning by Ant Colony Optimization. International Journal of Computer Applications in Technology, 12, 1999.
9. K. Bolding, M. L. Fulgham, and L. Snyder. The case for chaotic adaptive routing. Technical Report CSE-94-02-04, Department of Computer Science, University of Washington, Seattle, 1994.
10. E. Bonabeau, M. Dorigo, and G. Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999.
11. J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom98), 1998.
12. D. Camara and A.F. Loureiro. Gps/ant-like routing in ad hoc networks. Telecommunication Systems, 18(1–3):85–100, 2001.
13. H.S. Chang, W. Gutjahr, J. Yang, and S. Park. An Ant System approach to Markov Decision Processes. Technical Report 2003-10, Department of Statistics and Decision Support Systems, University of Vienna, Vienna, Austria, September 2003.
14. C. Cheng, R. Riley, S.P. Kumar, and J.J. Garcia-Luna-Aceves. A loop-free extended bellman-ford routing protocol without bouncing effect. ACM Computer Communication Review (SIGCOMM '89), 18 (4):224–236, 1989.
15. L. Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In Proceedings of the Tenth National Conference on Artificial Intelligence, pages 183–188, 1992.
16. I. Cidon, R. Rom, and Y. Shavitt. Multi-path routing combined with resource reservation. In IEEE INFOCOM'97, pages 92–100, 1997.
17. I. Cidon, R. Rom, and Y. Shavitt. Analysis of multi-path routing. IEEE/ACM Transactions on Networking, 7(6):885–896, 1999.
18. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL), 34:39–53, 1994.
19. M. Cottarelli and A. Gobbi. Estensioni dell'algoritmo formiche per il problema del commesso via

ggiatore. Master's thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, 1997.
20. J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. Journal of Insect Behavior, 3:159–168, 1990.
21. J.L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The dynamics of collective sorting: robot-like ants and ant-like robots. In S. Wilson, editor, Proceedings of the First International Conference on Simulation of Adaptive Behaviors: From Animals to Animats, pages 356–365. MIT Press, Cambridge, MA, USA, 1991.
22. G. Di Caro and M. Dorigo. AntNet: A mobile agents approach to adaptive routing. Technical Report 97–12, IRIDIA, University of Brussels, Belgium, June 1997.
23. G. Di Caro, F. Ducatelle, and L.M. Gambardella. AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks. In Proceedings of Parallel Problem Solving from Nature (PPSN) VIII, volume 3242 of Lecture Notes in Computer Science, pages 461–470. Springer-Verlag, 2004.
24. G. Di Caro, F. Ducatelle, and L.M. Gambardella. AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. European Transactions on Telecommunications, 16(5), October 2005.
25. E. W. Dijkstra. A note on two problems in connection with graphs. Numerical Mathematics, 1:269–271, 1959.
26. M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. Artificial Life, 5(2):137–172, 1999.
27. J.J. Garcia-Luna-Aceves and S. Murthy. A path-finding algorithm for loop-free routing. IEEE/ACM Transactions on Networking, February 1997.
28. F. Glover. Tabu search, part I. ORSA Journal on Computing, 1:190–206, 1989.
29. J. Moy. OSPF version 2. Request For Comments (RFC) 1583, Network Working Group, 1994.
30. J. Moy. OSPF Anatomy of an Internet Routing Protocol. Addison-Wesley, 1998.
31. G. Owen. Game Theory. Academic Press, third edition, 1995.
32. M.L. Puterman. Markov Decision Problems. John Wiley & Sons, 1994.
33. Rome, Kay, and Friedemann Mattern. The design space of wireless sensor networks. IEEE Wireless Communications, 11(6):54–61, December 2004.

This page is intentionally left blank