RED-f Routing Protocol for Complex Networks

Wilson Wang-Kit Thong and Guanrong Chen
Department of Electronic Engineering
City University of Hong Kong
Hong Kong SAR, China
Email: {wilsonwk, gchen}@ee.cityu.edu.hk

Ljiljana Trajković
School of Engineering Science
Simon Fraser University
Vancouver, British Columbia, Canada
Email: ljilja@cs.sfu.ca

***Abstract*—In this paper, we address routing in complex networks. Routing traffic across a network requires finding best possible paths between sources and destinations. When data traffic changes dynamically, a path that was optimal in the past may not be the best for the next packet. Adapting to traffic changes and finding optimal paths dynamically are challenging tasks. They become more demanding in large and complex networks.**

In optical burst switching (OBS) networks, two optical bursts contending for the same link need resolution mechanisms other than queueing. Deflection routing protocols are used to override routing tables and "deflect" one of the bursts to a free link. Instead of deflecting bursts at an immediate point of contention, the proposed Random Early Deflection (RED-f) routing protocol triggers deflection ahead of time and, thus, offers additional routing paths and lowers the burst loss rate due to contention. Simulations demonstrate that RED-f enabled nodes in a scale-free complex network reduce burst loss rate by exchanging control information with only few other network nodes.

I. Introduction

Routing protocols that respond to instantaneous traffic changes improve performance of communication networks for a wide range of applications. Examples include control systems that use networks in their feedback paths. Routing protocols of this type have been proposed and analyzed in statistical physics [1], [2], where phase transition [3] of a cloud of particles (data traffic) in a complex structure (network) is investigated and modeled. Mechanisms are suggested to manipulate clouds microscopically in order to attain certain macroscopic statistical properties. While the modeling methodology does not address engineering issues (design of signaling and protocol architecture), the proposed approach may be applied to emerging communication networks such as optical burst switching (OBS) networks [4].

In this paper, we propose the Random Early Deflection (RED-f) routing protocol. The protocol allows optical bursts (collections of packets) to circumvent busy links ahead of time and reduce overall burst loss rate. RED-f combines deflection routing [5] and Random Early Detection (RED) [6], [7] algorithms. We have implemented RED-f using the ns-3 network simulator [8]. Integrating RED-f module in ns-3 makes its software architecture compatible with protocol stacks in deployed data networks.

This research was supported by the NSERC Discovery Grant 216844-07.

In Section II, we briefly review literature related to routing algorithms in statistical physics and in communication networks. We describe details of the RED-f algorithm in Section III. We present RED-f simulation results in Section IV and conclude with Section V.

II. Routing Protocols

A. Communication Networks

A variety of routing technologies have been designed to address traffic and topology changes in a communication network.

Open Shortest Path First (OSPF) [9] is a de facto routing protocol in industry [10]. Paths are automatically recalculated if network topology changes. OSPF relies on the Dijkstra algorithm [11] to find minimum-cost paths between two nodes. Cost of links may be optimally configured to distribute traffic evenly over a network and minimize congestion spots [12].

Enhanced Interior Gateway Routing Protocol (EIGRP) is a vendor proprietary protocol. It is based on the Diffusing Update Algorithm [13]. When status of a link (up or down) or a link's weight changes, the protocol enables faster convergence of routing tables and guarantees absence of routing loops (even temporarily).

MultiProtocol Label Switching (MPLS) allows network operators to mark certain packets to be routed along predefined paths and is scalable to large networks [14]. Paths may be chosen arbitrarily and need not necessarily be least-cost choices. This flexibility enables planning backup paths for possible link or node failures. Instant or rapid failover mechanism becomes feasible [15]. Moreover, paths may be individually engineered to carry traffic loads optimally across a network [16].

The described algorithms react to network topology changes rather than traffic changes. As routing algorithms take time to converge, reacting to traffic changes on the packet-per-packet time scale is infeasible. However, shaping traffic on the finer time scale may be achieved by employing queuing management, admission control, RED [6], [7], or their combinations to build a service differentiation architecture [17].

B. Statistical Physics

Power law distribution of node degrees and scale-free properties have been discovered in various complex networks, such as social, biological, and communication networks. Studies in the area of statistical physics have explored possible causes

behind these common properties. Several network models have been proposed to explain the phenomena by reproducing empirical observations [18], [19].

Recently proposed improvements in designing routing protocols [1] rely on insights emanating from complex networks. The algorithm increases network throughput and reduces packet queuing delay by exchanging queue length information with neighboring nodes. Packets are routed to a next hop with higher probability if its queue is shorter.

Reacting to traffic loads during routing has also been proposed [2]. Instead of routing packets over paths with the least number of hops, packets are routed on paths with the least number of hops h and the least sum of node queue lengths c. The trade-off between the two criteria is controlled by weights. Packet transmission time is minimized by choosing a larger weight associated with queue length c.

These studies demonstrated possible performance improvements. However, they have not addressed signaling protocols and implementation issues.

III. RED-F: Random Early Deflection Protocol

RED-f protocol is designed for the OBS network architecture [20]. In OBS networks, multiple packets are grouped into one data burst and a burst control header (BCH) packet is created. The BCH packet contains information such as burst duration, source address, and destination address. It is sent ahead of its data burst, with an offset time t_{offset}. When an OBS node receives a BCH packet, it reads the destination address, searches for the corresponding outgoing interface, and configures its optical cross connect (OXC) module within the t_{offset} time period. Once the burst arrives, it traverses through the OXC and leaves for its next hop on the chosen outgoing interface [21].

The main difference between an OBS node and an electrical switch is the absence of first-in-first-out buffer to queue bursts because optical signals cannot be stored. Hence, bursts contending for the same output link require designing contention resolution schemes other than queuing. For simplicity, in this paper we consider deflection routing [5] as the only resolution scheme. Other schemes may also be implemented [4].

RED-f is based on early deflection and randomness. Early deflection is illustrated by considering the four possible routing paths from node n_s to node n_d shown in Fig. 1. Although there are four paths, blocking two paths at node n_x is sufficient to cause burst losses at the node. Furthermore, node n_y has no option to resolve burst contentions by deflection. Therefore, we propose early deflection where nodes are notified backward along paths when contention occurs. If links attached to n_x or n_y are busy, bursts may be deflected earlier at node n_α or even at node n_β. Additional paths then become available for deflection. Randomness is introduced to the early deflection in order to balance traffic among the four paths.

Operation of the RED-f protocol is illustrated in Fig. 2. When there is no burst contention, RED-f enabled nodes route bursts to outgoing interfaces according to the least-hop routing

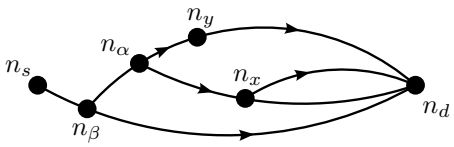

Fig. 1. Routing paths from node n_s to node n_d.

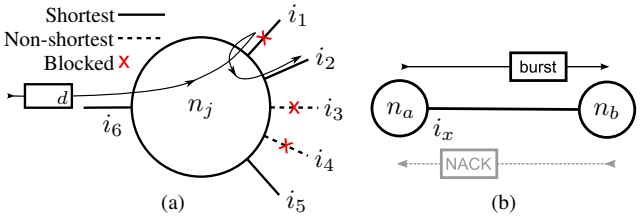

Fig. 2. (a) RED-f node n_j applies deflection on a burst addressed to destination d. (b) Illustration of the backward notification protocol.

(LHR). Various routing schemes may be combined with RED-f. We choose LHR for its simplicity. When there is contention, RED-f executes two actions: penalty-based deflection and backward notification.

1) Penalty-Based Deflection: A burst is deflected to the interface $i \in \mathcal{D}$ in a deflection set $\mathcal{D}$ according to its penalty $p(i, d)$, where d is the destination address of the burst.

Consider the burst at the transiting node n_j shown in Fig. 2(a). Interfaces i_1, i_2, and i_5 are on the least-hop paths toward destination d. Path i_1 is selected by LHR for forwarding the burst. However, i_1 is currently blocked because it is occupied by other bursts. RED-f then constructs a deflection set $\mathcal{D}$. The set $\mathcal{D}$ contains all the non-blocking interfaces (solid lines without crosses shown in Fig. 2(a)) laying on the least-hop paths toward d, thus $\mathcal{D} = \{i_2, i_5\}$. Larger set $\mathcal{D}$ increases the chance of successful deflection. Nevertheless, adding interfaces arbitrarily to $\mathcal{D}$ introduces risk of routing loops. Approaches for creating larger deflection sets without loops have been proposed [22].

A probability $b(i)$ is calculated from penalties p as $b(i) = 1 - \frac{p_i - p_{\text{thmin}}}{p_{\text{thmax}} - p_{\text{thmin}}}$, where $p_i = p(i, d)$ and p_{thmin} and p_{thmax} are two configurable parameters. RED-f randomly selects an interface for deflection. Each $i \in \mathcal{D}$ has $b(i) / \left(\sum_{k \in \mathcal{D}} b(k)\right)$ chance to be selected. For example, i_2 is selected in Fig. 2(a). The randomness allows balancing traffic over interfaces $i \in \mathcal{D}$ proportionally to $b(i)$. Because i_1 calls for a deflection, RED-f penalizes i_1 by setting

$$p(i_1, d) \leftarrow p(i_1, d) + 1.0. \tag{1}$$

Hence, subsequent bursts from node n_j toward destination d will use interface i_1 less often. The effect of the penalty p decays exponentially with time at rate α, which is a user configurable parameter. Hence,

$$p(t + \delta) = p(t)\mathbf{e}^{-\alpha\delta}. \tag{2}$$

If no burst contention occurs for a certain period of time, RED-f behaves as LHR.

2) Backward Notification: When a burst deflection occurs, a negative acknowledgment (NACK) is sent to upstream

TABLE I
SIMULATION PARAMETERS IN SIMPLE NETWORKS

Description	Value	Description	Value
Link bandwidth	1 Gbps	p_{thmax}	1.0
Link propagation delay	0.1 ms	p_{thmin}	0.1
Number of wavelengths per link	1	α	45
Burst size	125 kbytes	w	3 ms

nodes. A backward notification tree (BNT) keeps track of the interfaces that should be used for sending future NACKs. The BNT is a set $\mathcal{B}(i_o, d)$ of interfaces. When a burst addressed to d is blocked on an outgoing interface i_o, NACK is sent to every interface in $\mathcal{B}(i_o, d)$. When a burst leaves an interface i_o, an interface i_i is added to the $\mathcal{B}(i_o, d)$, where i_i and d are the incoming interface and the destination address of the burst, respectively. Every burst triggers adding an interface whether or not it is deflected. Shown in Fig. 2(a) is a burst addressed to d that arrives from i_6 and leaves at i_2. Hence, i_6 is added to $\mathcal{B}(i_2, d)$. In order to reduce NACK traffic, an interface added to a BNT remains in $\mathcal{B}(i_o, d)$ only for w seconds and it is removed when the timer expires. Parameter w is user configurable.

Temporary BNT interface and NACK notification act as a pair to define a protocol. Node n_a transmits a burst to node n_b, as shown in Fig. 2(b). If no NACK is returned within w seconds, n_a considers the transmission successful. Otherwise, n_a updates the penalty. When the burst is blocked at node n_b at time t_s, an NACK is sent from node n_b. The burst's destination address d and t_s are added to the NACK message. At the same time, node n_a *tries* to update the penalty $p = p_{n_a}(i_x, d)$ based on (1): $p(t_s) \leftarrow p(t_s) + 1.0$. However, the update may only be completed after the NACK arrives to n_a. Assuming that NACK arrives at time t_r, (2) mandates that the penalty p at time t_r decays as: $p(t_r) \leftarrow (p(t_s) + 1.0) \times \mathbf{e}^{-\alpha(t_r - t_s)}$. Expanding the right-hand side and using (2) leads to:

$$\begin{aligned} p(t_r) &\leftarrow p(t_s)\mathbf{e}^{-\alpha(t_r-t_s)} + 1.0 \times \mathbf{e}^{-\alpha(t_r-t_s)} \\ &= p(t_r) + 1.0 \times \mathbf{e}^{-\alpha(t_r-t_s)}. \end{aligned} \tag{3}$$

Note that the penalty is designed to decay exponentially (2). Equation (3) is the update algorithm used by node n_a when it receives an NACK. After n_a receives the NACK and updates the penalty, it sends the NACK upstream. The NACK travels backward hop-by-hop, subject to the existence of BNT interfaces.

IV. SIMULATION RESULTS

We evaluated performance of the proposed RED-f protocol using the ns-3 network simulator [8]. Ns-3 is an open source tool developed in C++ that features realistic implementations of network protocols. Details of ns-3 and comparison of its performance with other network simulation tools may be found in [23].

A. RED-f Performance in Simple Networks

We first simulated the RED-f protocol using a simple network topology with two flows f_1 and f_2 that compete for the link n_c–n_e, as shown in Fig. 3. RED-f deflects bursts in

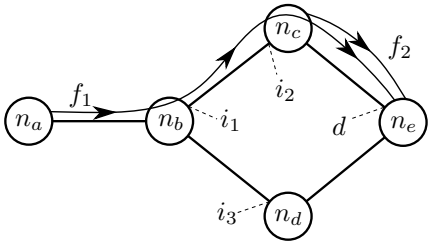

Fig. 3. RED-f performance in a simple network. Labeled are four interfaces: i_1, i_2, i_3, and d.

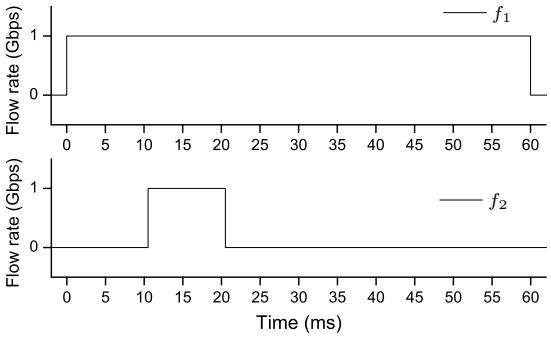

Fig. 4. Transmission periods for flows f_1 and f_2.

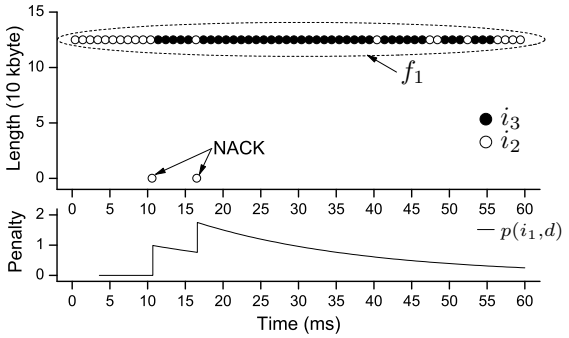

Fig. 5. Bursts captured at interfaces i_2 and i_3 shown in Fig. 3 (top). Penalty $p(i_1, d)$ at n_b is used for routing bursts toward d (bottom).

f_1 at n_b to avoid contention although the contention occurs at n_c. Simulation parameters are shown in Table I. Flows f_1 and f_2 send 60 and 10 bursts, respectively, as shown in Fig. 4. Contentions occur on link n_c–n_e between 10 ms and 20 ms. Flows f_1 and f_2 send traffic at constant rate of 1 Gbps.

RED-f avoids the contention between flows f_1 and f_2, as illustrated in Fig. 5. Each dot in Fig. 5(top) represents a burst captured on interface i_2 or i_3. The burst length is shown on the y-axis. Before flow f_2 becomes active, no contention occurs and f_1's bursts travel along the preferred path. Thus, the bursts appear only on interface i_2. When f_2 begins emitting bursts at 10 ms, contentions occur on link n_c–n_e. An NACK (18 bytes) is generated and sent from node n_c to node n_b. The penalty $p(i_1, d)$ at node n_b is shown in Fig. 5(bottom). When n_b receives the NACK from n_c at 10 ms, the penalty is increased by 1.0. Bursts in flow f_1 then immediately change the path at n_b, as shown in Fig. 5(top). The bursts are only seen on interface i_3 after the increment of the penalty.

As the penalty decays exponentially after receiving the NACK, the probability of using the original f_1's path increases

TABLE II
NUMBER OF BURSTS TRANSMITTED (TX) AND RECEIVED (RX)

Tx	Rx (LHR)	Rx (RED-f)
70	60	68

TABLE III
SIMULATION PARAMETERS IN SCALE-FREE NETWORKS

Description	Value	Description	Value
Number of nodes	1,000	Number of bursts	4×10^5
Number of links	1,996	Number of simulation runs	10
Number of flows	2,000		

accordingly. At approximately 15 ms, one f_1's burst is routed back to the original path, leading to the second burst contention with flow f_2. Another NACK is sent from n_c to n_b and the $p(i_1, d)$ is increased again by 1.0. After the second burst contention, the penalty becomes so large that f_1's bursts change their path at node n_b for a period of time (at least up to 40 ms) longer than the period after the first contention, as shown in Fig. 5. As the penalty continues to decay, f_1's bursts gradually change the path back to normal. Further burst contentions do not occur because flow f_2 is already terminated. With RED-f, 8 out of 10 bursts in flow f_2 have been recovered. The number of bursts received at node n_e with or without RED-f are shown in Table II.

B. RED-f Performance in Complex Networks

We also simulated RED-f performance in larger scale-free networks [18]. Simulation parameters are listed in Table I and Table III.

Burst loss rate is plotted as a function of the total rate of traffic injected into the network, as shown in Fig. 6. On average, RED-f exhibits approximately half the burst loss of LHR for traffic rates below 10 Gbps. The extent of improvement is limited by the number of the least-hop paths between nodes. Since most nodes are of degree 2, the improvement achieved by RED-f is reasonable. Note that the $1/2$ difference is statistically significant for data points < 10 Gbps as confirmed by the analysis of variance (not shown here) [24]. Further improvements in burst loss rate is possible by constructing a larger penalty-based deflection set $\mathcal{D}$ [22]. RED-f burst loss rate may be smaller in denser networks with larger average node degrees.

V. CONCLUSIONS

Based on results from statistical physics, which demonstrated the advantages of reacting to traffic dynamics when routing, we have proposed and designed a new RED-f routing protocol. The protocol has been simulated using the ns-3 network simulator. Its performance has been evaluated by considering various network topologies. Future enhancements may include optimizing RED-f parameters and considering the effect of network topology on the protocol performance.

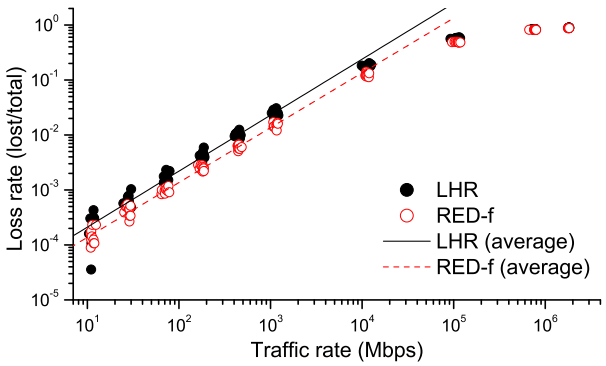

Fig. 6. RED-f performance in a scale-free network.

REFERENCES

[1] W.-X. Wang, C.-Y. Yin, G. Yan, and B.-H. Wang, "Integrating local static and dynamic information for routing traffic," *Phys. Rev. E*, vol. 74, p. 016101, July 2006.

[2] P. Echenique, J. G.-Gardeñes, and Y. Moreno, "Improved routing strategies for Internet traffic delivery," *Phys. Rev. E*, vol. 70, p. 056105, Nov. 2004.

[3] G. Grimmett, *Percolation*, 2nd Ed., Ser. *Grundlehren der Mathematischen Wissenschaften*. Berlin, Heidelberg, Germany: Springer-Verlag, 1999, vol. 321.

[4] L. Xu, H. G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Commun. Mag.*, vol. 39, no. 1, pp. 136–142, Jan. 2001.

[5] F. Borgonovo, *Deflection Routing*. Hertfordshire, UK: Prentice Hall, 1995, Ch. 9.

[6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[7] S. Floyd. (1997). *RED: Discussions of Setting Parameters* [Online]. Available: http://icir.org/floyd/REDparameters.txt.

[8] *Ns-3 Network Simulator* [Online]. Available: http://www.nsnam.org/.

[9] J. T. Moy, *OSPF Version 2*, IETF RFC 232, Apr. 1998.

[10] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Boston, MA: Addison-Wesley, Feb. 1998.

[11] T. H. Cormen, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

[12] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 4, pp. 756–767, May 2002.

[13] B. Albrightson, J. G.-L.-Aceves, and J. Boyle, "EIGRP–a fast routing protocol based on distance vectors," in *Proc. Networld/Interop 94*, 1994.

[14] A. Viswanathan, N. Feldman, Z. Wang, and R. Callon, "Evolution of multiprotocol label switching," *IEEE Commun. Mag.*, vol. 36, no. 5, pp. 165–173, May 1998.

[15] J. L. Marzo, E. Calle, C. Scoglio, and T. Anjah, "QoS online routing and MPLS multilevel protection: a survey," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 126–132, Oct. 2003.

[16] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 42–47, Dec. 1999.

[17] J. Evans and C. Filsfils, *Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice*. San Francisco, CA: Elsevier Science, 2007.

[18] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[19] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, Jan. 2002.

[20] Y. Chen, J. S. Turner, and P.-F. Mo, 'Optimal burst scheduling in optical burst switched networks," *J. Lightw. Technol.*, vol. 25, no. 8, pp. 1883–1894, Aug. 2007.

[21] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "Jumpstart: a just-in-time signaling architecture for WDM burst-switched networks," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 82–89, Feb. 2002.

[22] X. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," in *Proc. ACM SIGCOMM'06*, New York, NY, USA, Oct. 2006, pp. 159–170.

[23] E. Weingärtner, H. Vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *Proc. IEEE ICC'09*, Dresden, Germany, June 2009, pp. 1287–1291.

[24] R. R. Johnson and P. J. Kuby, *Elementary Statistics*. Pacific Grove, CA: Duxbury Press, 2006.