Extending Dynamic Reconfiguration to NOWs with Adaptive Routing*

Aurelio Bermúdez\(^{2}\), Francisco J. Alfaro\(^{1}\), Rafael Casado\(^{2}\), José Duato\(^{3}\),
Francisco J. Quiles\(^{2}\), and José L. Sánchez\(^{2}\)

\(^{1}\) Department of Computer Engineering and Technology
University of Murcia, 30071- Murcia, Spain
faifaro@eltecn.un.es

\(^{2}\) Department of Computer Science
University of Castilla-La Mancha
02071- Albacete, Spain
\{abermu,rcasado,paco,jsanchez\}@info-ab.uclm.es

\(^{3}\) Department of Information Systems and Computer Architecture
Technical University of Valencia
46071- Valencia, Spain
jduato@gap.upv.es

Abstract. Many distributed applications executed on networks of work-
stations (NOWs) require the interconnection network to provide some
quality of service (QoS) support. These networks must be able to sup-
port topology changes (due to component failures, hot replacement, hot
expansion, etc.) without stopping traffic, in order to satisfy QoS require-
ments. Traditional network reconfiguration methods do not take this into
account, causing a serious performance degradation while the network is
being reconfigured.

In [1, 2], we proposed a new dynamic network reconfiguration protocol,
called Partial Progressive Reconfiguration. It significantly reduces the
negative effects produced by traditional methods. For this reason, it is
especially suitable for applications requiring QoS. This reconfiguration
protocol requires that messages are routed using up*/down* routing.

In this paper, we extend this dynamic reconfiguration technique to sup-
port adaptive routing, based on the design methodology for adaptive
algorithms proposed in [3, 4]. We also present performance evaluation
results, clearly showing the benefits of using dynamic reconfiguration
combined with adaptive routing.

1 Introduction

NOWs are usually composed of several hosts interconnected by means of switches
and point-to-point links in an arbitrary topology. Network topology may change

\* This work was partly supported by the Spanish CICYT under Grant TIC97-0897-C04, and Caja Castilla-La Mancha
due to switches and hosts being turned on/off, link remapping, component failures, hot replacement, hot expansion, etc. In these cases, a reconfiguration process must update the routing tables to guarantee network connectivity. Several current reconfiguration algorithms need to stop user traffic during reconfiguration to prevent deadlock. For this reason, this methodology is called static reconfiguration. Autonet [5] and Myrinet [6] are two representative examples.

Several distributed real-time applications have strict communications requirements [7, 8], with rigorous limitations on CDV (cell delay variation), CTD (maximum cell transfer delay) and CLR (cell loss ratio) parameters. Distributed multimedia applications have similar, although less strict, quality of service (QoS) requirements. Nowadays, many distributed multimedia applications such as real-time video compression and decompression, video-on-demand servers, distributed databases, etc., require computing power beyond that available in current uniprocessors. These applications require very high network bandwidth, which can be provided by means of a high-speed LAN.

When multimedia applications are executed on a local area switch-based network, topology changes may affect their behavior. If static reconfiguration is used, user traffic is stopped and the average packet latency increases dramatically during the reconfiguration. It should be noted that traffic is stopped in the entire network, even if topology changes only affect a small region of the network. Thus, it will not be possible to guarantee the required QoS [9] during reconfiguration. Moreover, messages buffered in their source nodes during reconfiguration will be transmitted immediately after finishing the reconfiguration, possibly leading to network saturation and preventing QoS from being guaranteed for even longer. Stopping user traffic has even worse effects on distributed real-time applications due to the more strict timing constraints required for them.

A different approach to solve this problem is dynamic reconfiguration. Unlike static reconfiguration protocols, dynamic protocols do not require stopping network traffic during the reconfiguration in order to guarantee the absence of deadlock. This fact leads to shorter packet delays and reduced packet loss rates even during the reconfiguration. This approach is applicable to networks that attempt to provide delivery guarantees and QoS. Note that dynamic reconfiguration by itself does not guarantee QoS. However, it is not possible to guarantee QoS without dynamic reconfiguration when the topology changes.

A dynamic reconfiguration protocol, called Partial Progressive Reconfiguration (PPR), was proposed in [1, 2]. This approach avoids deadlock during reconfiguration by applying a sequence of partial routing table updates. PPR was designed for virtual cut-through networks without virtual channels and distributed up*/down* routing. On the other hand, networks with virtual channels can support adaptive routing, which is much more efficient than up*/down* routing. In fact, adaptive routing can use up*/down* routing to escape from deadlock [3, 4]. This work focuses on extending dynamic reconfiguration to NOWs that support virtual channels and adaptive routing.

The next section presents up*/down* routing and describes a methodology to extend it to adaptive routing. Also, both static and dynamic reconfiguration