TOTEM: A FAULT-TOLERANT MULTICAST GROUP COMMUNICATION SYSTEM

L. E. Moser, P. M. Melliar Smith, A. Agarwal, B. K. Budhia, C. A. Lingley-Papadopoulos
University of California, Santa Barbara

Communications of the ACM, 1996

1. Introduction
The Totem system provides reliable totally ordered multicasting of messages over LANs.

2. Totem Services
Totem is built as a hierarchy of protocols:

<table>
<thead>
<tr>
<th>Application Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Group Interface</td>
</tr>
<tr>
<td>Multiple-Ring Protocol</td>
</tr>
<tr>
<td>Single-Ring Protocol</td>
</tr>
<tr>
<td>Physical Medium</td>
</tr>
</tbody>
</table>

The bottom layer of this hierarchy is a best-effort multicast service typically using UDP to exploit the hardware broadcasts of the LAN. The single ring protocol converts these multicasts into the service of reliable totally ordered delivery of messages on a single LAN. It also provides fault-detection, recovery and configuration change services. The multiple ring protocol, using information from the process group interface above it provides total ordering of messages as well as network topology maintenance services. The process group interface delivers messages to the application processes in the appropriate process groups and provides process group membership services.

Totem provides two reliable totally ordered message delivery services:

a) Agreed delivery, which guarantees that a processor will not deliver a message before it has delivered all prior messages that have been issued by processors in the current configuration and have time-stamps within the duration of that configuration;

b) Safe delivery, which further guarantees that a processor will not deliver a message unless all processors in its configurations have received it (everyone or nobody).

Both services deliver messages in a single system-wide total order that respects Lamport’s causal order. As a result, processes belonging to two or more groups will always receive messages from different groups in the same order.

Totem uses extended virtual synchrony to ensure that the above agreed and safe delivery guarantees are honored within each group even if faulty processors are repaired or if a partitioned network remerges.

To achieve extended virtual synchrony, Totem uses born-ordered messages: the total order of messages can be determined directly from the messages.

3. The Totem Single-Ring Protocol
The single-ring protocol uses a circulating token containing, among other things:

a) a seq field containing the sequence number of the last message that was sent,
b) an aru field containing the sequence number of the last message that has been received by all processors: it is used to implement safe delivery.

Only the processor that holds the token can send a message. The aru field allows processors to purge from their sending buffers the messages that have been received by every processor in the ring. The token also provides information about the aggregate message backlog of the processors on the ring, allowing a fairer bandwidth allocation among processors than simpler schemes such as FDDI.

The local membership protocol allows inclusion of new or recovering processors and deletion of faulty processors. It ensures:

a) consensus among all members of a configuration about the configuration membership, and

b) termination, in the sense that each configuration will be installed on every processor within a bounded time or not at all.

4. The Totem Multiple-Ring Protocol
Breaking a ring into several rings will often increase throughput. The multiple-ring protocol provides the same services as single-ring protocol.

To achieve a global total order of messages over all rings, the Totem multiple-ring protocol uses Lampart’s timestamps and delivers messages in timestamp order. When a gateway forwards a message from one ring to another, it gives to the message a new sequence number for the new ring.

For each ring from which it can receive messages, each processor maintains a list of messages that were received but not yet delivered (recv_msgs). A processor will deliver a message as an agreed message as soon as this message has the lowest timestamp of all the messages in its recvmsgs lists and none of these lists is empty. Since this last condition could cause long delays, guaranteed vector messages specifying, among other things, which rings have sent messages are sent from time to time.

Processor faults and network partitions are detected by the single-ring protocol. Configuration and topology change messages have timestamps and are delivered in strict timestamp order along with regular messages.