Broadcast to a nodegroup results in the message being copied once for each destination node. These are then sent in a loop.

This results in numnodes copies of the message, which has a time, space, and bandwidth cost proportional to the size of the message and the number of hosts.

For large messages (where large varies by machine layer, but 128k should be fine) this should be replaced by an RDMA scheme that arranges for each destination to RDMA get the payload. Thereby reducing the space and time cost. This could be further enriched by a spanning tree approach to reduce the single link bandwidth cost, though at the price of increased latency at the leaves. Probably best to cut over to that scheme when the number of hosts is large, and/or the size of the message is very large (~1G). This should be implementable using our existing zero copy semantics to avoid writing a new implementation at each machine layer. If not, then we should extend our semantics to facilitate this kind of usage.

An even more advanced scheme would use message layer broadcast primitives. However, their applicability and portability has its own research agenda, so a distinct subtask should be created for that by whoever goes that direction.