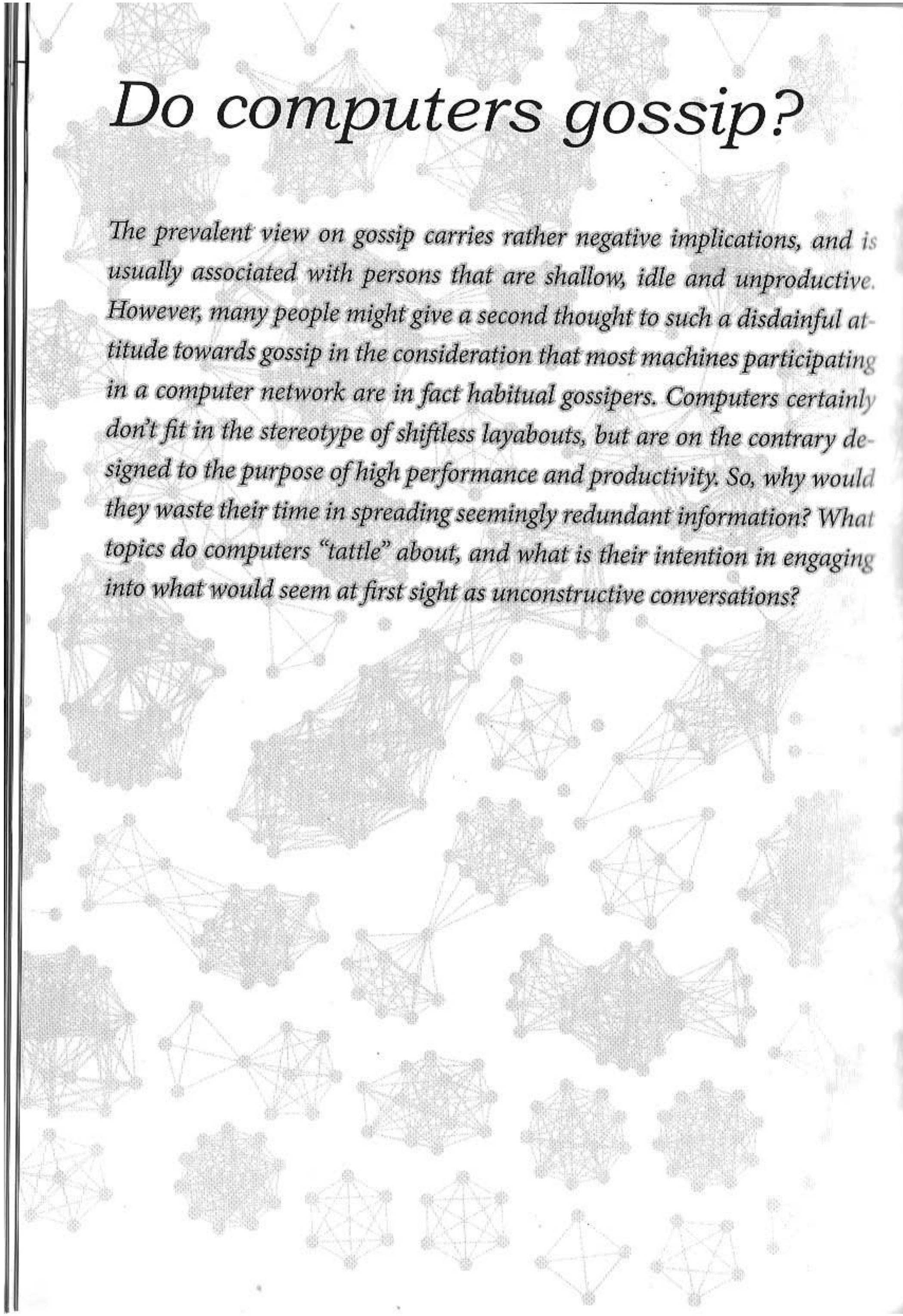

Do computers gossip?

The prevalent view on gossip carries rather negative implications, and is usually associated with persons that are shallow, idle and unproductive. However, many people might give a second thought to such a disdainful attitude towards gossip in the consideration that most machines participating in a computer network are in fact habitual gossipers. Computers certainly don't fit in the stereotype of shiftless layabouts, but are on the contrary designed to the purpose of high performance and productivity. So, why would they waste their time in spreading seemingly redundant information? What topics do computers "tattle" about, and what is their intention in engaging into what would seem at first sight as unconstructive conversations?

Setting aside the obvious hurtful consequences of speaking ill of another person or exposing sensitive details of one's private affairs, recent research in disciplines such as anthropology and evolutionary biology has shown that gossiping plays an essential role in forming and sustaining the bonds of large, dynamic social groups.

As the size of a social network increases and relations between its members change constantly, knowledge based on eye-witnessing is limited to a small fraction of the population, and communication can no longer be grounded on direct connection links. Thus, abundant information exchange and reliance on second-hand observations and opinions becomes indispensable for reinforcing the cohesion and cooperation level of the group.

In the past years, the size and complexity of computer networks has evolved dramatically, leading to the establishment of powerful distributed systems, where multiple autonomous computers -called nodes or peers- are linked together over e.g. some wireless or telephone network, and interact by passing messages to each other. Each computer has its own private local memory, where it stores its data, and can only find out about the current state of remote nodes by consulting its peers for relevant information. Like in social groups, each node has only a partial view of the overall system, and thus the design of robust communication protocols is a paramount demand for efficient coordination.

Given such analogies between the organisation of human communities and the architecture of computer networks, the adoption of patterns from the theory of social networking, and especially of functions attributed to gossipping, can prove very helpful to the design of scalable and reliable distributed systems.

The fundamental characteristic underpinning gossip, which is a common experience in everyday life, is that it provides a quite simple but potentially very efficient mechanism for the fast spreading of news. In distributed systems, "hot" news may e.g. concern the recently observed operational status of a communication link, report some new membership or the unreliable behaviour of a peer, inform about updates in the local database of a node, include some request, or advertise a service provision. By maintaining a rather "verbose" attitude, and being willing to engage in conversations at a regular rate, nodes can keep themselves up-to-date about the latest developments in their network. As is the case with human beings, a sense of reciprocity is underlying this disposition to circulate abundant information: gossipy nodes are eager to propagate their recent discoveries, expecting that its peers will return the favour.

In this context, the selection of a set of gossip partners, with whom a node exchanges data, is crucial for achieving robust information dissemination. Each node running a gossip protocol periodically picks a subset of its neighbours, i.e. the computers it is directly linked with, sends to them the data it wants to share - concerning its own state or something it has recently learnt about other nodes - , and collects the information known to its gossip peers in return. Communication partners are selected in a random-like fashion, so that at every round of the protocol the nodes engaged to the gossip interaction are renewed with some probability, and thus it is impossible for a node to build an isolated cluster of gossips. This way, the spreading of information has been shown to be particularly fast, and to scale very well with network size. Another valuable property of this unpredictable selection pattern is that it makes gossip-based algorithms resilient to modifications in the network topology and message losses. If a communication link breaks, or a node crashes, broadcasted information is still highly likely to find its way through some alternative route. Thus, throughput, i.e. the average rate of successful message delivery, remains notably stable and reliable even under the presence of failures and continuous change.

On the other hand, if nodes select their gossip partners in a purely random basis and trust blindly whatever information they receive, gossiping becomes vulnerable towards nodes with "unsocial" behaviour, that only care about maximising their own benefit or even worse have the malicious intention to abuse the system. A selfish node, for example, that is interested in the knowledge possessed by some specific peers of it, will try to gossip only with these peers, to learn as early as possible about their updates, while being reluctant to forward the knowledge it possesses. Such a node is obviously a bad choice for being a gossip partner. To make things nastier, a node may be malevolent enough to proactively pursue the detriment of other nodes, e.g. by deliberately initiating an excessive number of useless gossip interactions, and imposing arbitrary load to benign nodes. Moreover, a malicious process might take advantage of the gossip operations to diffuse false rumors. To mitigate the harmful impact of unfaithful nodes, computers have to be careful in their peer selection, by regulating its random nature, and propagate observations about deviating behaviours. By maintaining black lists of uncooperative-cooperative partners, nodes can refuse gossip requests from processes that have been spotted as unreliable. Cryptographic schemes, such as digital signatures, can be used to authenticate the source of a message and ensure that the message was not altered while in transit.

Another potential drawback of gossip, that may undermine the performance of a distributed system, stems from the very characteristic that makes it so robust, namely the extravagance in information transfer, by having nodes propagating probably redundant information at fixed frequent intervals. If too much redundancy is introduced, the available bandwidth may be reduced, leading to high dissemination delays and compromising throughput. Therefore, it is important to balance the rate of information flooding according to the capacity of the system, and avoid as much as possible sending data to nodes that already know about it.

From the above examples follows that, just as in real life, the harmful implications of gossip result from the presence of untrustworthy members, who manipulate its mechanism to promote only their own interests or contaminate the reputation of others, as well as from the overindulgence in its use. However, if such behaviours are collectively deterred, gossip can act as a vector for building a persistent sense of community with shared interests and information. It may sound peculiar, but modern computer networks are probably the most convincing advocates of such a positive function of gossip, and the most expert practitioners of its rather defamed art.

Eirini Kaldeli

Distributed Systems

FURTHER READING

Gossip-based computer networking. ACM SIGOPS Operating Systems Review, Volume 41, Issue 5, 2007.

BAR Gossip. H.C. Li et al. In Proc. USENIX Operating Systems Design and Implementation (OSDI), 2006.