Bandwidth-efficient delay- and loss-tolerant overlay routing

Wojciech Galuba, Karl Aberer
Ecole Polytechnique Fédérale de Lausanne (EPFL)
wojciech.galuba@epfl.ch
project: http://lsirpeople.epfl.ch/galuba/ffp

Zoran Despotovic, Wolfgang Kellerer
DOCOMO Euro-Labs, Munich, Germany

Motivation. Peer-to-peer (P2P) systems are mostly deployed in heterogeneous environments with resource availability varying not only across the nodes but also over time. If any of the shared computational, storage or network resources are exhausted, failures and delays occur. The commonly used crash-stop failure model assumes that once a node stops sending messages it never again resumes. Such failures are trivially detected and appropriate algorithms are run that maintain the connectivity and routing efficiency of the P2P overlay under continuous arrivals and departures of the peers (i.e. churn) [6], [4].

The failure detection mechanisms in the crash-stop model are typically tuned to minimize the number of false positives that might be caused by intermittent message dropping or delays. Avoiding these false positives is important as oversensitive failure detection triggers more overlay maintenance events. Overlay maintenance is costly, not only because the peers need to run the necessary protocols for acquiring new neighbors, but also because the applications (e.g. DHTs) using the overlay need to respond to the failures as well. For these reasons the handling of the non-permanent failures cannot simply be delegated to the overlay maintenance. These failures may significantly affect overlay routing and additional fault-tolerance mechanisms are necessary.

The failure model. The causes of message loss and delays can be numerous. In heterogenous P2P systems running ever more intensive workloads nodes may become overloaded [5]. Networks may experience transient connectivity problems [3]. External adversaries can mount DDoS attacks [2], while the internal adversaries can take control over a fraction of the peers in the system and disrupt the message passing protocols [1].

In this paper we abstract away from the causes of failures and subsume them in a well defined failure model. A fraction of peers are allowed to arbitrarily delay or drop messages. The drops and delays can occur in a message-dependent way. However, we forbid message mutation and spurious message injection.

The protocol. Within the above failure model we address the problem of reliable recursive message routing in structured overlays. In our **Forward Feedback Protocol (FFP)** each routed message is followed on its routing path by a feedback message. Feedback signals either success or failure of message delivery. Peers accumulate feedback and based on it adjust their routing decisions. Routing path delays exceeding a timeout and dropped messages trigger negative feedback, which leads to readjustment of the paths to route around the peers causing delays or loss.

Each peer locally keeps a set of *success estimators* for each of its neighbors. The success estimators are random variables reflecting the history of the past routing outcomes. When a message arrives and needs to be forwarded the peer draws samples from the success estimators. Based on these samples the peer probabilistically picks the next hop that maximizes routing success. When the feedback message subsequently arrives it is used to update the success estimators. Over time as the peer is forwarding service requests and receiving feedback it improves its routing decisions.

The proposed FFP protocol has the following properties:

- **path-wide fault-tolerance** - FFP's failure detection covers the whole routing process: from the moment the source sends the message, until the final destination acknowledges the receipt. Thanks to that FFP can respond to failures that can only be detected at the routing path level, such as when the delay accumulated over the whole path exceeds the application timeouts.

- **low overhead** - peers in our system continuously gather the routing performance data about their neighbors and use it for making the routing decisions. This is in contrast to some of the existing approaches which rely on multiple redundant paths for increasing fault-tolerance. Evaluation shows that despite the 2-5 times lower bandwidth usage our solution achieves the success rate comparable to the existing approaches.
- **zero-knowledge routing** - FFP peers do not require any prior knowledge about their neighbors, even their location in the ID space. Through feedback each peer individually learns which of its neighbors are reliable forwarders for which destinations and the network as a whole converges on efficient routing paths.
- **scalability** - FFP is fully decentralized and scalable, we show the local state size to be only $O(\log^2(N))$ in terms of the network size.
- **universality** - FFP is general enough to be used with any recursively routing overlay or in general in any recursively routing network.

Evaluation. The system is implemented using the ProtoPeer[1] toolkit and evaluated in a PlanetLab deployment.

We consider the following routing protocols: (1) **NM** - no routing fault-tolerance mechanisms, (2) **MULTIk** - multipath routing as in [1], in the first hop the source chooses k neighbors closest to the destination instead of one, (3) **ITERk** - iterative routing scheme based on Kademlia[4] with k simultaneous lookups and (4) **FFP** - system running the FFP. In all setups we use a bidirectional Chord [6] implementation. Each peer sends one service request to a random destination ID every 500-1500ms. Each deployment is approximately 350 PlanetLab hosts in size. We are using real-world Kademlia churn traces taken from [7]. FFP's resilience to the dropping attacks is comparable to that of the existing approaches (Fig. 1) while at the same time FFP has 2-5 times lower bandwidth consumption (Fig. 2). FFP also detects and routes around peers excessively delaying messages and tolerates churn.

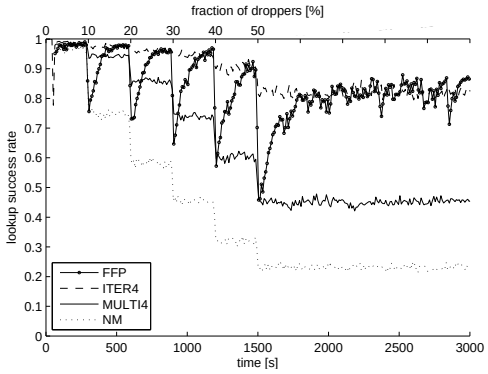

Fig. 1. **Resilience to message dropping.** Every 5mins. a new 10% batch of peers starts to drop all messages except their own lookup traffic. FFP rapidly responds to the failures and routes around the droppers and reaches performance level of the existing approaches.

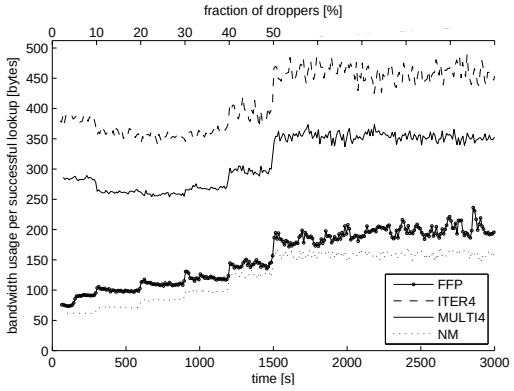

Fig. 2. **Bandwidth usage per message delivery.** Setup as in Fig. 1. Iterative and multipath routing relies on 4-way message redundancy for fault-tolerance, which results in high bandwidth usage. FFP's only overhead are the feedback messages, which are small in size and can be sent in bulk.

REFERENCES

[1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure routing for structured peer-to-peer overlay networks," *ACM Operating Systems Review*, vol. 36, no. si, p. 299, 2002.

[2] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, and W. Zwaenepoel, "Denial-of-service resilience in peer-to-peer file sharing systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 38–49, 2005.

[3] M. J. Freedman, K. Lakshminarayanan, S. Rhea, and I. Stoica, "Non-transitive connectivity and dhts," in *WORLDS'05*. Berkeley, CA, USA: USENIX Association, 2005, pp. 10–10.

[4] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *IPTPS*, ser. LNCS, vol. 2429. Springer, 2002, pp. 53–65.

[5] S. Rhea, B. Chun, J. Kubiatowicz, and S. Shenker, "Fixing the embarrassing slowness of opendht on planetlab," 2005.

[6] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *SIGCOMM'04*, 2001, pp. 149–160.

[7] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *IMC'06*, J. M. Almeida, V. A. F. Almeida, and P. Barford, Eds. ACM, 2006, pp. 189–202.

[1] http://protopeer.epfl.ch/