Remote Measurement of NNTP Internet Traffic

T. Surmacz

Wroclaw University of Technology,
Institute of Computer Engineering, Control and Robotics
Email: tsurmacz@ict.pwr.wroc.pl

Abstract. *Measuring internet traffic related to NNTP protocol (Usenet News) provides vital information needed for optimum news server configuration. However, gathering data is usually restricted to the "local view", even though in networked environment an overview of the neighbourhood traffic would be more appropriate. This article describes method of gathering such data from remote points and analysing it to provide the required traffic flow information.*

Keywords: Measuring Internet Traffic, NNTP Protocol, Usenet News

1. Introduction

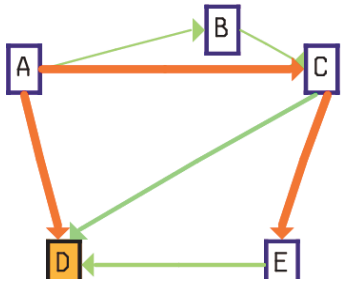
Usenet is a large network of servers exchanging messages (articles) sent by users throughout the world in various discussion groups. Message sent on any server in a particular discussion group gets propagated to all other servers carrying this group through a flooding algorithm [1] that provides a good tradeoff in achieving two main purposes: delivering the messages to all servers and saving the required bandwidth by not transmitting the message to servers already containing it. However, as full usenet news traffic often exceeds 200-300 GigaBytes of data per day on a single server-to-server link, and a server may have several such links open to other servers, more sophisticated tools are required to properly engineer such a traffic, especially that there are usually many redundant links between servers set up to increase the reliability of the overall system.

Measurement of this traffic is the first step for optimizing network bandwidth usage of a news server. The typical setup, however, is to gather locally generated statistical data, which gives only a "local view" of the news server operation. This may be used to spot irregularities or trends in day-to-day operation, but does not provide a "bigger picture" of the overall news traffic in the neighbourhood of servers. In order to get this bigger picture, acquisition of traffic data from remote servers is required.

2. Subject of Measurement

Let's consider a simplified network of just 5 servers located in two different cities (Fig. 1). Servers A, B and C are connected by a wide bandwidth metropolitan area network, as well as servers D and E sharing a fast connection, but the link between these two groups is much slower. If a message is posted to server A (either by a local user, or received from some external link), it will be propagated to all neighbouring servers (B, C and D). The same happens all over again on every consecutive server, so server B will try to propagate it to C, while C will try to push it through to D and E. Servers D and E will eventually exchange the message too, depending on which of them receives it first, but there is a good chance, that the message will traverse the slow link twice (from A to D and from C to E) instead of being distributed mostly by fast links. A way of news bandwidth control by message diverting has been proposed in [2], however a proper measurement method to evaluate the results is still required.

Fig. 1. Sample news servers network topology



Considering the message propagation shown in Fig. 1, server D can calculate the incoming traffic from all its neighbours, but cannot determine the bandwidth used between them (e.g. the traffic between C and E or B and C). Even if data collection is possible from nodes other than D, it will not be possible for all nodes, so some information must be derived from the data collected at other points of the network.

The aim of a measurement method described in this article is to estimate the bandwidth used between all the servers being considered, by analysing the data which is already available locally in headers of received messages, and possibly retrieving the missing data from other servers.

3. Measurement Method

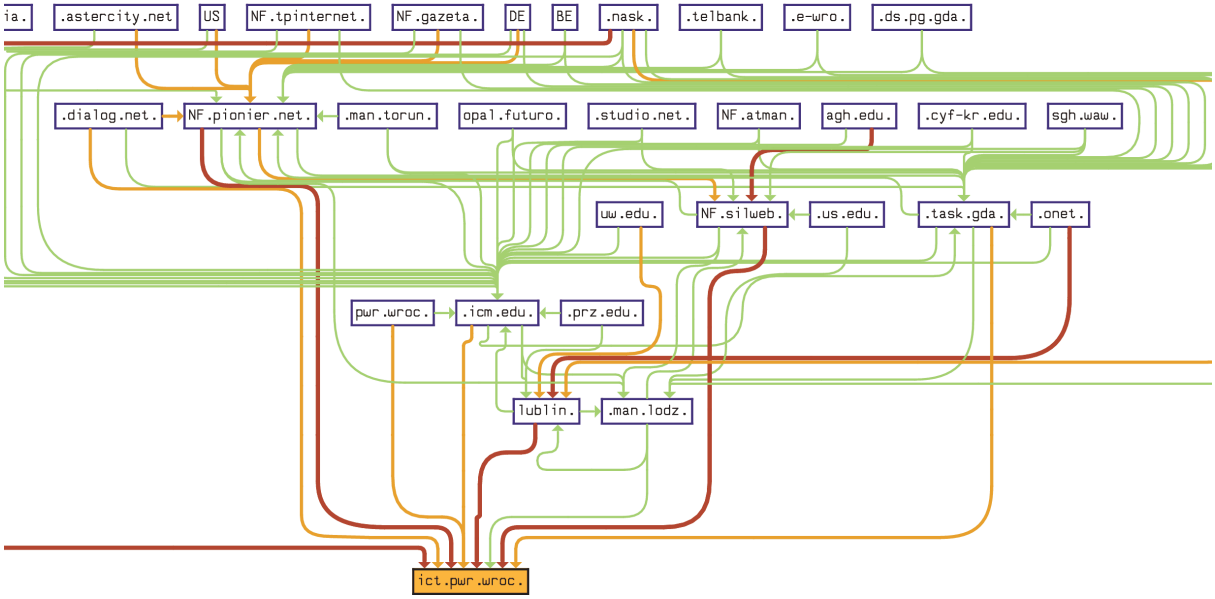
Messages sent through the Usenet News system contain two main parts: a header and a body. The header contains all the information about the message origin (author, date posted, server used for posting, etc.), as well as the “Path:” field, which accumulates the information about the servers the message has been sent through. So if a message is sent from server A to B, then from B to C, the Path: field will contain “Path: C!B!A”, as it shows all server names in reverse order, split by the exclamation mark. This data is normally used by NNTP servers to reduce the required bandwidth by not offering the message to servers that have already seen it (i.e. server C will not offer this article to B nor A), but may also be used to calculate the NNTP-related bandwidth of remote links.

The method for calculating the bandwidth first uses all the locally contained messages and analyses them by splitting the “Path:” field. The directed graph is being built, in which nodes represent servers and edges represent server links. Each message coming from server A through servers B and C to server D (i.e. “Path: D!C!B!A”) adds to edges AB, BC and CD the size of the message (called its *weight*) which is known, as the message is stored locally. However, if the Path: header contains only vectors BA and AD, then it is not known how the message was delivered to server B, or how server C has been offered this, if at all, and to which other edges should the appropriate weight be added. Another target of measurement may be to count just the number of messages sent, but this is a trivial simplification done by ignoring the actual size of messages.

The next step is to analyse the data that can be obtained from remote servers. If other servers offer public access to their repositories, it is possible to connect to them using the NNTP protocol as a client and retrieve message headers for analysis. This must be done in several steps:

1. connect to a remote server as a client;
2. retrieve the list of accessible groups;
3. retrieve the numbers of the first and the last message in each observed group;
4. compare these numbers to the results of a previous session with this server;
5. retrieve headers of all messages that have not been seen in previous run and store them locally.

Fig. 2. Visualisation of Path analysis results for server ict.pwr.wroc.pl, based on its local data only. For picture clarity, the leading "*news*" and trailing "*pl*" parts have been omitted in server names and such names are left with a starting or ending dot. The "*newsfeed*" prefix in real names has been replaced with "*NF*".



Steps 3-5 have to be repeated for each discussion group and steps 1-5 for each server that can be used for getting reading access. It is possible to retrieve only some of the existing header fields using XHDR and XOVER NNTP protocol extensions described in [3]. The headers used in this method are "Path:", "Xref:", "Message-ID:", "NNTP-Posting-Date:", "Date:" and "Bytes:". The "Xref:" header contains the list of groups where the message has been posted and may be used to speed up data processing by simplifying message exclusion algorithms (so it does not get counted twice if sent to two different groups). The two posting dates (one set by the client, the other by the server) are currently not used, but may be included in propagation delay analysis or for selecting messages sent in particular period (for now, this is done by collecting data in regular intervals and analysing them in batches). The most important headers are the remaining ones – "Path:", "Bytes:" and "Message-ID:". The latter one contains a unique identifier of the message which can be used to correctly identify all messages whose weights (the "Bytes:" header) have been already added to particular graph edges and should not be counted again on these links.

Consider the last example – the message that comes from server B to A, and then from server A is sent independently to servers C and D. When it is analysed on D as a locally stored message, the Path: header contains "Path: D!A!B", so its weight is added to vectors BA and AD. The header of the same message retrieved from server C would contain "Path: C!A!B", so its weight would get counted twice on edge BA. To prevent that, it is necessary not only to add message weights, but also to keep track of their Message-IDs in a sorted list associated with each edge being considered.

4. Results

The results of the counting algorithm are twofold – first, it allows to find NNTP connections between remote servers and draw a graph showing them, second – it finds the bandwidth used

in these connections. Analysis performed on data gathered lately (beginning of February 2009) showed surprisingly large number of redundant links between servers to the extent which made creating any visual graphs impossible, as they were not readable at all, with links spanning in all directions. Fig. 2 shows the results of stage 1 "Path:" analysis from local data, gathered on server ict.pwr.wroc.pl. The weights calculated for particular links (represented by line thickness) had to be used to exclude all links below some threshold values in order to make this graph readable. However, for actual calculations of server "importance" and the bandwidth it utilizes, all the edge weights should be used. Line thickness/color represents the importance of the link, i.e. the number of messages being sent. Fig. 3 shows the bandwidth analysis results of remotely gathered data, presented in graph form, and restricted to traffic in "pl.*" subset of discussion groups only. For graph clarity, all edges and nodes with less then 10 MB of traffic have been omitted and the numbers represent kilobytes of data sent. Highlighted nodes are the servers used for data collection. The analysed data was collected between 7.02.1009 and 14.02.2009 and consisted of 322477 article headers (71236 unique articles) with 258 different data paths discovered before the graph reduction.

5. Discussion

The counting algorithm presented in this paper helps system administrators of NNTP servers to visualize how messages are being sent between the servers and calculate the bandwidth used for NNTP service between local and remote servers. This data may be used to optimise network usage by modifying server setup and divert traffic between existing server links helping to achieve both the required reliability from redundant links, and efficiency. The nature of data collection implies that the counted numbers cannot be 100% accurate, as this would require connecting to every single server on the network and including its data in overall analysis, which would be impractical, if possible at all. However, by Message-ID: analysis performed in parallel to weight counting, it is possible to find what messages have been omitted in calculations for particular node, and thus – estimate the corrections for edge weights related to this node. This will be marginal for nodes for which the data was collected using the NNTP protocol, but may be significant for those implied from Path: analysis and graph building. However, for "important" servers which are large message exchange hubs these uncertainty numbers are also low, as the messages sent through them are widely distributed through other servers, and thus – are included in the overall analysis.

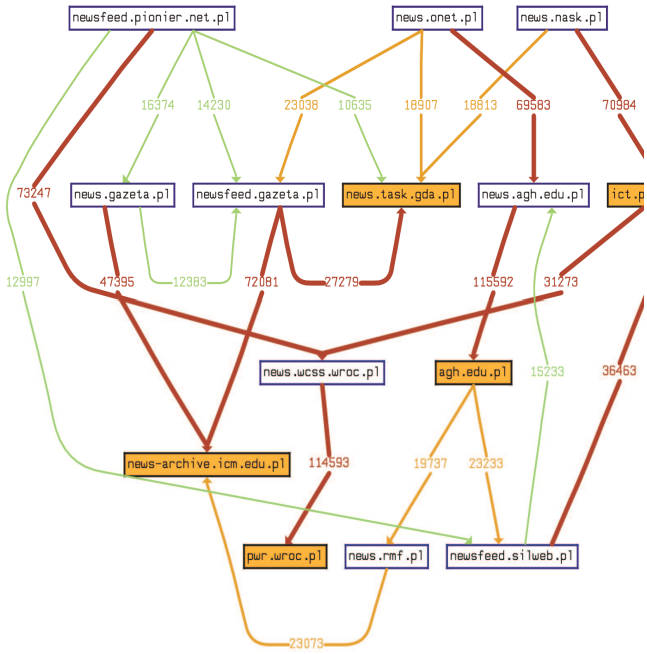


Fig. 3. Full analysis results – kilobytes of messages sent between servers

6. Conclusions

The method described in this article allows measurement of NNTP traffic in IP networks based on just few probe points. The results of these measurements may be used by Usenet news system administrators to monitor daily bandwidth usage and assist in traffic engineering, or calculate various characteristics of news flow between news servers.

References

[1] Horton M, Adams R. Standard for interchange of USENET messages. *RFC 1036.* Dec 1987.

[2] Surmacz T. Bandwidth control in redundant news server links. In proceedings of International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX 2006), Szklarska Poręba, Poland, IEEE Computer Society 2006, 143-149.

[3] Barber S. Common NNTP Extensions. *RFC 2980,* Oct 2000