Measurement of data flow in usenet news management

Tomasz R. Surmacz

Institute of Engineering Cybernetics, Wroclaw University of Technology
ul. Janiszewskiego 11-17, 50-372 Wroclaw, Poland
email: tsurmacz@ict.pwr.wroc.pl

Abstract. *Usenet news is a worldwide network of servers providing discussion forums with topics ranging from scientific research to hobbies, entertainment or politics. Current volume of news stream can take up to 200 gigabytes per day if binary groups are included, so appropriate tools for measuring data flow between servers are needed to allow the optimal usage of the underlaying network bandwidth. The paper describes methods of measuring the flow of articles, delays introduced by message transmission and other data, as well as problems of detecting anomalies and dealing with them in order to assure a smooth operation of the whole system. Conclusions are drawn from the data gathered over the last 12 months.*

Keywords: usenet news, data flow measurement

1. Introduction

Discussion forums are one of the key elements of modern electronic communications. A worldwide network of servers called "Usenet news" [1] is one of the oldest, but still the most popular such a system, allowing computer users all over the world to share their opinions on various topics. There are several thousand groups (such as "*comp.software.measurement*") organised in hierarchies (eg. "*comp.**", "*sci.**", "*rec.**", etc.) Articles are sent by users to a nearest news server of the local Internet Service Provider (ISP) and are automatically distributed throughout the whole web of cooperating news servers. An article sent from any of the servers flows through the whole network from one server to another (via a link called a newsfeed), leaving exactly one copy at each cooperating server. Appropriate mechanisms exist to prevent data from looping when multiple connections exist.

Problems associated with running a news server [2] include:

- maintaining redundant links to other news servers, in order to provide reliable incoming news flow and to allow propagation of locally posted articles to the world;
- monitoring those links for a possible network congestion and detecting failures or unannounced configuration changes at remote sites;
- monitoring data flow statistics for traffic exceeding some predefined thresholds (meaning usually some Denial of Service (DoS) attack or a misconfigured server somewhere in the net).

Usually, when a server failure occurs, such as an overflown news spool (no free disk space) or a dried news feed (lack of new articles at all), it is already too late to start finding the reasons, unless some reference data has been gathered during the normal operation of the news server.

2. Subject and purpose of measurement

Several quantities can be measured in a running news server system. Articles are being constantly received, forming a massive stream of incoming traffic, but this stream can be easily classified on a per-article basis, as each article can be logged with additional data, such as:

S – article size in bytes;
N – newsgroup name (or a list of newsgroup names) in which the article appeared;
F – the news server from which the article was received;
O – names of all news servers (outgoing peers) the article will be sent to;
T_p – time and date the article was originally posted;
T_r – time and date the article was received;

All these values are worth considering only when summarized or compared over some period of time, eg. gathered as hourly or daily statistics. Also, much more interesting then the plain sum of the article sizes are the results of processing data selected by other factors, eg. number of articles in some hierarchy or a total volume of articles sent to a particular server.

From the maintenance point of view, the most interesting values are:

- Average article delay, measured as a difference between T_r and T_p;
- Incoming flows – volume and count of incoming articles per site (newsfeed) or per hierarchy;
- Outgoing flows – acceptance rate and volume of outgoing newsfeeds.

Measuring article delays can help detecting backlogs of data accumulating at remote ends of incoming newsfeeds. If such backlogs exist, they usually mean an insufficient bandwidth of the underlying network or some other efficiency problems between two news servers. Solving that usually requires redesigning the structure of incoming newsfeeds.

Keeping day-to-day statistics about incoming flows can help identifying problems whenever a sudden change of volume or number of articles occurs. A sudden increase of data usually means some news system abuse, such as posting binary data in discussion groups (eg. ripped CDs or pirated copies of programs), increased level of SPAM, or a Denial of Servic e attack.

Outgoing flows should be monitored for potential backlogs (meaning again the available bandwidth is exhausted) but also for clues how to maximize efficiency of all the newsfeed links (incoming and outgoing).

3. Methods

Delay between sending an article at some remote news server and receiving it locally can be calculated by comparing T_r and T_p. However, the value of T_p depends on the clock accuracy of a remote server used to post the article and cannot be always trusted to be correct. In fact, quite often the timezone of a news server appears to be set incorrectly, giving a bias in multiples of 60 minutes. This can be easily detected if measured article transfer time yields negative values, but otherwise it is not detectable in a reliable way.

Sample of such a measurement is shown in fig. 1. Data on the left describes articles "posted in the future", but as 90% of them falls under 7 minutes, it simply means a slight clock skew at remote sites. Two little peaks at -1h and +1h mean probably an incorrect timezone or a wrong daylight saving time correction set at remote servers (resulting in 1h difference from the official time). Other peaks at +12h and +1 day mean probably an off-by-one error when setting the date at some servers and possible AM-PM error possible in countries using 12-hour clock.

Delays can be calculated in two different ways: online – by monitoring T_p and T_r logged by news server as new articles are being received, or offline – by analysing a so-called *history file* (containing article IDs and their T_r values) and finding each article in disk spool area to get the Tp value from article headers. This may fail if an article has already been expired or cancelled (ie. deleted from news spool).

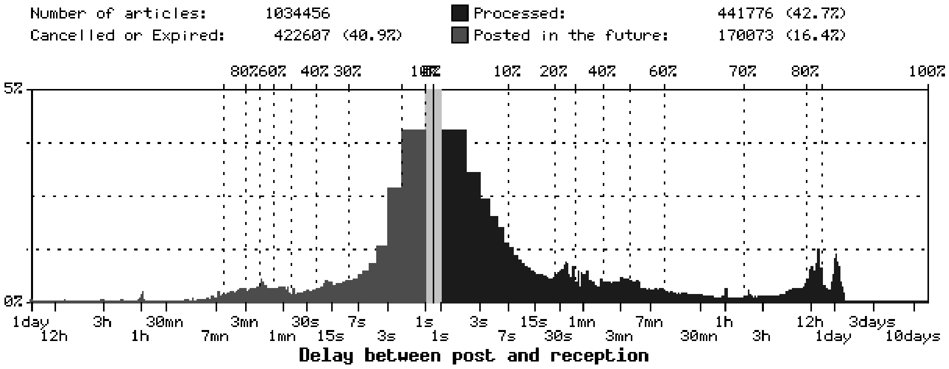

Fig. 1. Delay in reception of articles.

For measuring input and output data flows, a simple one-time setup is usually required. As new articles come, the news server writes a special log file, recording all the required information: the article size (S), newsgroup name (N), time received (T_r), and in/out feed names (F, O). Then a separate process can analise this data periodically, creating hourly and daily summaries and snapshots of system state.

Incoming volume and count of articles can be split by incoming newsfeed name, showing how many articles come from each neighbouring news server. A graphical representation of these data over a longer period of time is desired, to spot problems such as a disappearing stream of articles from one server, a sudden increase of volume in one of the newsfeeds, or a shift from one newsfeed to another. None of these requires immediate action, if proper redundant links are maintained, unless the changed situation persists. Fig. 2 shows sample of such data. At around –100 hours (counted from plot generation time) one can observe that newsfeed.pionier.net.pl stops sending articles, but more of them comes from news.task.gda.pl at the same time, meaning a temporary problem at "pionier" and a proper operation of redundant links. At –78 hours there is a peak in volume of articles received from "pionier". When compared to another plot, showing article count for that time (not shown here) with no anomalies at all, one can draw a conclusion, that there must have been some large articles posted at that time (eg. binary postings) and if disk space was exhausted, appropriate log files can be checked to see which news groups or hierarchies these articles were sent to.

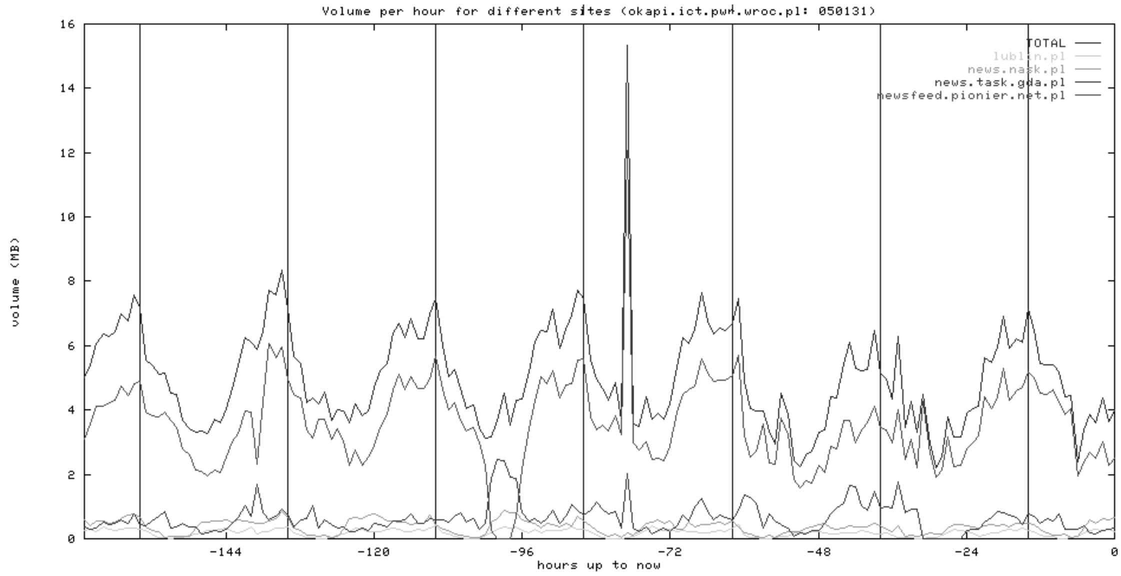

Fig. 2. Incoming flow of articles split by newsfeed.

Analysis of flow data can also help reconfiguring incoming and outgoing data streams to utilize the bandwidth of the underlying network links in a best way.

Let's consider a transit server B connected with two other servers A and C. Each article sent from A is stored locally and retransmitted to C, and vice versa. B and C are located at the same city (connected through a high bandwidth metropolitan area network) and the main direction of news flow is from A through B to C. If, for increased reliability, C has some other connections, the goal to achieve is the fastest possible propagation of articles between B and C, as the articles received at C over a local high speed network will not be transmitted again from remote sites over much slower long-distance links.

To prevent network bandwidth waste, an incoming news volume per server should be monitored as well as outgoing news volume and rejection rate to other servers. Eg. if high volume of news coming from A to B is accompanied by a 90-100% acceptance rate of articles sent from B to C, the bandwidth is utilized properly (based on their unique identifiers, C will refuse to accept them again from other servers before the actual transmission). However, poor acceptance rate between B and C as well as high volume of data on A-B links means that the same articles travel multiple times through slow long-distance links and some newsfeed reconfiguration is needed.

4. Results

Based on data gathered during years 2003-2004, article delays on a fully functional news servers usually do not exceed several minutes. Ie. 50% of articles sent anywhere in the world will be available for reading at any other news server within few minutes and 90% of them – within one hour.

The flow data in numerical form can be used to automatically raise alarms when some predefined thresholds are exceeded, eg. resulting in disk space problems or bandwidth overloading. However, except some trivial examples, such fixed thresholds are usually hard to set, as they depend on many factors, such as the network topology and available bandwidth, acceptable delays, desired reliability, allowed link redundancy, and sometimes even local policies for exchanging news – so for a better view of the running system a visualisation of gathered data is preferred. Such graphical data can be presented on WWW pages, helping system administrators to maintain a proper operation of the news system.

5. Conclusions

Constant monitoring is needed to maintain the usenet news system running smoothly. Measuring the right type of data and proper analysis of these data are the key to identifying potential problems and solving them. Analysis of existing article flows can be used to redesign server connections and techniques such as deploying some minor delays in article sending can help optimising usage of available network bandwidth. The ongoing process of measuring data flow at news.ict.pwr.wroc.pl news server can be seen at http://www.usenet.pl/stat/.

References

[1] Kantor B., Lapsley P. Network News Transfer Protocol: A proposed standard for the stream-based transmission of news, Feb 1986. RFC977.

[2] O'Reilly T., Todino G. Managing UUCP and Usenet. O'Reilly & Associates, Inc., March 1988.