

Time Synchronization in DCNET Hosts
D.L. Mills, COMSAT Laboratories
25-Feb-81

Introduction

The difficulty in establishing an absolute time reference for use in internet measurements and experiments has been often lamented. While time standards calibrated to the precision necessary for internet delay measurements (in the order of a millisecond) are readily available, the cost to equip each host which may participate in these experiments is significant and the broadcast services they depend upon may not be available where needed. This note describes an alternative mechanism using local-net protocols to synchronize a logical clock in each of a set of internet hosts to a single physical clock, such as an NBS radio clock. The mechanism has been incorporated as an integral component of the DCNET network routing algorithm and depends for its accuracy upon the careful control of link delays. For this reason it may not be practically applicable as a retrofit in the ARPANET, for example. Nevertheless, the principles can be applied in cases where somewhat less precision is acceptable and where the participating hosts or gateways support the required protocol.

The DCNET Routing Algorithm

The DCNET includes a number of PDP11-compatible hosts interconnected by dedicated and dial-up links of various types, including simple synchronous and asynchronous point-to-point links, high-speed interprocessor channels and self-contained retransmission systems. All of these links include inherent delays which vary with transmission rate, message length and coding, as well as occasional retransmissions. In addition, the host operating system can introduce delays due to interrupt latency, interprocess communication and process scheduling. These delays typically include a fixed component due to propagation phenomena, together with a relatively small variable component due to internal queueing, coding and retransmission mechanisms.

The DCNET architectural design includes the notion of virtual host, which is a process resident in a physical host and labelled with a unique internet address. One or more of these virtual hosts can reside in a single physical host and can migrate about the network from time to time in arbitrary ways. Each virtual host can support multiple internet protocols, connections and, in addition, a virtual clock. Each physical host contains a physical clock which can operate at an arbitrary rate and, in addition, a 32-bit

logical clock which operates at 1000 Hz. Not all physical hosts implement the full 32-bit precision; however, in such cases the resolution of the logical clock may be somewhat less.

Routing of datagrams from a physical host to each of the virtual hosts in the network is determined by a set of Host Tables, one in each physical host, which are updated by HELLO messages exchanged on the links connecting them. The HELLO messages are exchanged frequently, but not so as to materially degrade the throughput of the link for ordinary data messages. They contain information necessary to compute the roundtrip delay and logical clock offset of the receiving physical host relative to the sending one, together with a table of delay and offset estimates computed between the sending physical host and each of the virtual hosts in the network. For the purpose of these estimates the delay and offset of each virtual host relative to the physical host in which it resides is assumed zero.

The Host Table is updated by HELLO messages from each neighboring physical host and in certain other cases. The updating algorithm is similar to that used in the ARPANET and other places in that the roundtrip delay calculated to a neighbor is added to each of the delay estimates given in its HELLO message and compared with the corresponding delay estimates in the Host Table. If a delay computed in this way is less than the delay already in the Host Table, the routing to the corresponding virtual host is changed accordingly. The detailed operation of this algorithm, which includes provisions for host up-down logic and loop suppression, is summarized elsewhere [1].

Virtual Clocks

The Host Table update procedure represents a convenient mechanism to implement a common time reference for each logical clock in the network. For this purpose each virtual clock residing in a physical host is assumed to run in synchronism with zero offset relative to the logical clock of that host. The offsets of the other virtual hosts in the network, relative to this logical clock, are computed along with the delays as HELLO messages arrive from neighboring physical hosts. A physical host, upon receiving a HELLO message, adds one-half the difference between its logical clock and its neighbor's logical clock contained in the HELLO message to each of the offset values in the message and stores the result in its own Host Table. Note that both the delay and offset values are stored only if the neighbor is in fact on the least-delay path to the virtual host and that the link delay is assumed equal in both directions. Also, note that should a virtual host move from one physical host to another, the delays and offsets in all Host Tables

relative to that virtual host would likely change.

Any user process in a physical host can reference its time-of-day calculations to any virtual host in the network by simply adding the appropriate virtual clock offset to the current value of the logical clock. Ordinarily, all network experiments use the same virtual host for this purpose, although not necessarily the same one in successive experiments. In the current implementation one of the physical hosts contains a special virtual host connected to an NBS radio clock. The offset stored in the Host Table corresponding to this virtual host reflects the offset of this clock relative to the logical clock. Using the above mechanism, the remaining physical hosts can reference their logical clocks to the receiver as well. One of the physical hosts is shortly to be moved not far from an atomic clock which is referenced to the Naval Observatory clock using a local television station. We are now assembling interface hardware for access to this standard and plan to use the above mechanism to reference all clocks to it.

Implementation Considerations

The absolute accuracy of the available NBS radio clocks is claimed to be of the order of a millisecond. Since this precision compares with that of the standard internet timestamps used for the most precise delay measurements, it would be natural to strive for a corresponding precision within a set of hosts using the mechanism described above. Ordinarily, this would require a crystal oscillator, counter and interface at each host. The oscillator stability found in commercial equipment of this type is of the order of a few parts per million under laboratory conditions. An uncorrected logical clock based on this equipment could be expected to maintain time to within a millisecond for at least eight minutes and to within three minutes in a year.

In the case of the mechanism described above, the corrections take the form of offsets contained in periodic HELLO messages. A careful analysis of the non-systematic errors inherent in these messages reveals contributions from a number of sources dominated by the following:

1. The interval between the instant the local clock is read and the departure of the first bit of the timestamp on the link.
2. The effect of the data coding conventions (e.g. character stuffing or bit stuffing) used to maintain data transparency.
3. The effect of retransmissions, where used.

4. The interval between the arrival of the first bit of the timestamp and the instant it is compared with the local clock.

The effects of the first and last of these delays has been minimized in the DCNET implementation by careful control of internal latencies and scheduling mechanisms and is limited to the order of a millisecond. The effects of character and bit stuffing can be estimated under the assumption that all character codes are equally likely. In that case, the effects of character stuffing would contribute a factor of about $1/256$ to the uncertainty in data rate and the effects of bit stuffing would contribute about $1/32$. Thus, for the case of 1200-bps character-stuffing links carrying HELLO messages of typically 800 bits, the uncertainty would be of the order of two milliseconds. The effects of bit stuffing in the high-speed links are negligible in comparison. Although HELLO messages are never retransmitted by a DCNET host, some of the DCNET links, in particular those based on the ACC Error Control Units (ECU), contain internal retransmission features. Since retransmissions occur so seldom in the present configuration, their effects have been ignored; however, a simple range-gate technique similar to that used in radar systems could be used to filter out retransmissions, should that become a problem.

From the above considerations the uncertainty in delays measured using HELLO messages can be conservatively assumed in the order of five milliseconds. For ease of analysis in the following, we will assume the uncertainty to be a random variable with a zero-mean Gaussian distribution and a standard deviation of five milliseconds. Thus, in order to reduce the uncertainty to the order of a millisecond, something over 25 independent samples would be required. The maximum interval between successive HELLO messages is thirty seconds, so that the required precision can in principle be achieved in about twelve minutes.

Précise determination of clock offsets requires that the drift rates between the various logical clocks be estimable with sufficient precision. The approach chosen in the DCNET design has been to initialize a variable representing the current offset of the local logical clock relative to that of a neighbor each time a HELLO message arrives from that neighbor which updates the Host Table entry for the selected virtual host. Once each second the period of the logical clock is increased by a quantity equal to $1/1024$ of this variable and the variable is decreased by the same quantity. The effect is that of a first-order recursive filter in smoothing the corrections and distributing them so as to minimize the phase jitter as viewed by the user process. In addition, if updates from the selected virtual host are lost due to failure of some host or link, the logical clock will

continue the correction process until the last offset received is compensated.

In order to provide for the initial setting of the logical clock and subsequent drift correction without disruptive phase discontinuities, the full 32-bit clock value is stored only if the (signed) offset exceeds 16 bits; that is, only if the high-order 16 bits must be changed. In other cases the low-order 16 bits are corrected by slewing the phase of the clock according to the current offset as described above. The high-order 16 bits correspond roughly to minutes, while the low-order 16 bits correspond to milliseconds. Most internet measurements will be concerned primarily with the latter, so this behavior is appropriate. This yields a slew rate of about one microsecond per second for each millisecond of offset.

The dynamics of this procedure insure a smooth transition in apparent drift rate from a maximum of 30 parts per thousand for the maximum offset to one part per million for the smallest. The maximum time required to slew the phase of a physical clock over the full (plus-minus) thirty-second range to steady state is about two hours. During the slewing interval the offset estimates continue to be valid, although of somewhat degraded accuracy.

In a multiple-host network where the logical-clock corrections must pass through a number of physical hosts, the robustness of this method depends on the cooperation of all intervening hosts. In general, this requires all hosts to track the same virtual host offset and, in addition, introduces additional dynamics in the drift-correction process. The effect is that of a set of coupled first-order recursive filters, where the input of each stage is connected to the output of the previous stage, and all have the same time constant described previously. So long as the drift rates are constant over at least several hours, this is not a problem; however, in the case where some logical clocks are derived from power-line sources, this can lead to significant loss in accuracy.

On Power Line Clocks

The short-term drift rates of power-line clocks relative to standard time have been observed to exceed several parts per thousand, with sharp changes in sign and magnitude occurring near periods of large load fluctuations. In experiments with the current DCNET implementation, discrepancies of several seconds are routinely observed between the power-line clock and the NBS radio clock. However, it is evident that the power systems for considerable portions of this country are closely synchronized in phase relative to each other. For instance, the DCNET hosts at COMSAT

Headquarters in Washington, D.C., and at Ford Scientific Research Laboratories in Dearborn, Michigan, have never been observed to slip a power-line cycle relative to each other.

These observations suggest that, if logical clocks are to be synchronized to an atomic clock or NBS radio clock, then physical clocks based on the power mains are not feasible, at least for accuracies of the order of milliseconds. On the other hand, for the short-term delay measurements required for many internet experiments and where reference to absolute time is not essential, the use of the power mains as a synchronization reference may be quite practical. This holds, of course, only in those cases where the power systems in the areas where measurements are made are in fact synchronous and probably would not apply, for example, for European sites.

On Radio Clocks

Two of the NBS radio clocks on the market include the Spectracom 8170 WWVB Synchronized Clock and the True Time 468-DC Satellite Synchronized Clock. The former uses the 60-KHz NBS broadcast from Boulder, Colorado, while the latter uses the NOAA GOES satellites. Both have claimed accuracy in the order of a millisecond and both support a service area including the continental US. A Spectracom 8170 is now in service connected to one of the DCNET hosts and serves as a master clock for the network. A few notes on its characteristics may be of interest, especially to others in the internet community planning on using similar equipment.

The characteristics of electromagnetic wave propagation at 60 KHz combine some features of the waveguide model applicable at frequencies below about 30 KHz and the ray model applicable at higher frequencies [2]. Both models explain how the E layer, an ionized region about 100 km above the earth's surface, guides or reflects electromagnetic waves over long distances. In the case of 60-KHz transmissions from Boulder, the models predict greater signal attenuation at times of the most intense ionizing radiation from the sun on the D region, which lies just below the E region and through which the signals must pass. Thus, one would expect that received signal levels would be highest at nighttime in winter and lowest in daytime in summer, with respect to the midpoint of the path from Boulder to the receiver. This has been observed in Washington, DC, where the receiver often drops out of phase lock for varying periods in the late afternoon. Measurements of the received signal level with a communications receiver confirm variations of 20-30 decibels.

Atmospheric discharges (called sferics) due to lightning can be a severe problem at these frequencies in Summer and could be expected to disrupt the receiver during summer evenings when propagation conditions are relatively good from active electrical areas like the Gulf Coast. Although other stations share the 60-KHz assigned frequency, including MSF in Rugby, UK, co-channel interference does not seem to be a problem. On the other hand, some household electrical appliances, including television deflection circuits and solid-state lamp dimmers, can generate copious harmonics

that disrupt the receiver. The ferrite antenna supplied with the receiver does not seem to be as effective as would ordinarily be expected in dealing with this problem.

Even when phase lock has been lost, the receiver coasts indefinitely using the last clock update received. Although the receiver indicates a signal loss of over ten minutes, both by a front-panel indicator and in the time message sent to the attached host, the indicated time should remain accurate to within an order of one part in a million, as suggested by tests using an accurate frequency counter. The DCNET implementation polls the receiver every thirty seconds and tracks the time messages as long as they are available, but retains the last received message for later inspection, if desired. In addition, if the time messages are lost the system continues to follow the information in the last message.

Summary and Conclusions

The above analysis indicates that logical clocks in neighboring physical hosts can be synchronized using ordinary local-network routing update messages, so long as the oscillator drift rates are stable and do not differ radically. A multiple-host network can readily be synchronized so long as each pair of neighboring hosts can, although this can require rather long settling times. Here, two hosts are assumed synchronized when the offsets between their respective logical clocks can be calculated to a precision of the order of a millisecond.

Delay and offset variances along internet paths are likely to be so large as to make the technique described above impractical, at least if millisecond accuracy is to be preserved. The present plan of installing radio clocks in the internet gateways appears to be the most sensible alternative. The gateways thus form an attractive time reference for the hosts on the attached local networks. The gateways themselves could include the offsets between their own clocks in Gateway-Gateway Protocol (GGP) routing update messages. If not all of them were equipped with RBS radio clocks, and could conveniently provide this information in GGP echo messages to the local network hosts. In order for

