The Token Grid Network

Terence D. Todd, *Member, IEEE*

Abstract—**An overwhelming majority of Local and Metropolitan Area Network products (LAN's and MAN's) are based upon linear topologies such as buses and rings [1], [2], [4], [9]. Such networks are economical for high speed operation since the station interfaces are simple and require very little transit buffering. However because of their linear structure, the total throughput is restricted by the transmission rate of the media access channels.**

In this paper, a token grid network is introduced where media access is performed over a two-dimensional mesh. In the resulting system, each station is two-connected and has the same transmission hardware and small station latency as in a dual token ring [3], [4]. In the token grid however, the total system throughput may be many factors larger than that which is possible in a dual token ring. In a large $\sqrt{N} \times \sqrt{N}$ network, the uniform load capacity is approximately $\sqrt{N}/2$ times that of an N station dual token ring. In addition, the token grid can take advantage of communities-of-interest amongst the stations. It is possible to implement the system in such a way as to achieve robust operation in the presence of station and link failures.

I. Introduction

MOST existing LAN/MAN products and standards are based upon linear shared media designs such as buses (e.g., IEEE 802.3 [1]) and rings (e.g., IEEE 802.5 and FDDI [2], [4]). These topologies are popular because it is possible to build a very simple and economical station attachment unit. In Ethernet for example, stations passively tap a broadcast bus medium and thus no high speed transit buffering is needed at the station. Similarly in a ring, the station interface need only buffer a small number of bits clocked at the network rate. These simple designs, coupled with the ease of controlling a linear network has led to the proliferation of many inexpensive products.

Due to the linear topology of rings and buses, the maximum throughput is severely restricted by the data rate of the shared channel. In a conventional token ring such as IEEE 802.5 [2], for example, the total throughput of the network is upper bounded by the channel data rate regardless of the number of stations. In principle, this problem may be addressed using a network topology which permits multiple concurrent packet transmissions. However, in many proposed designs the station simplicity of a bus or ring network must be sacrificed to achieve the improved performance properties. For example, in multihop or bridged networks [5], [8], high speed transit buffering and routing must be provided at each station. Although deflection routing [6] may be used to limit the buffering component, resequencing buffers are then required to recover from packet misordering due to deflections. In both cases the required station resources may be much higher than that in typical ring and bus networks.

In this paper, a network referred to as the token grid is introduced. The token grid is a multidimensional extension of the token ring [2], [4] where stations share access to a mesh formed by a set of overlapping rings. Media access is performed by having a station capture one out of a number of tokens circulating within the network. Couplings between the rings are implemented by the user stations in a very simple fashion and under token control. As in a token ring, the network supports the transmission of variable length packets. In the results presented, a dual token ring is used as a basis for performance comparisons with the token grid. This is because unlike other proposed networks, the station transmission hardware requirements for the token grid are identical to that of a dual ring network such as FDDI [4], [3]. Each station in both networks has exactly two transmitters and two receivers. Also, both networks may be implemented with a very small station latency where only a few bits per station need be buffered. Thus it is expected that the token grid design may be very cost-effective. In addition, the total throughput of the token grid may be many factors higher than that associated with a conventional dual ring or bus network [4], [9].

The paper is organized as follows. Section II provides an introduction to the network topology. This is followed by a discussion of the basic token grid protocol in Section III. In Section IV the performance of the token grid is considered. We include a brief discussion of network maintenance and reliability in Section V. In Section VI some concluding remarks are made.

II. The Token Grid Network

The token grid is a two dimensional network structure arranged in R rows and C columns. An example for $R = C = 4$ is shown in Fig. 1. Each station is identified by its row/column index as shown in the diagram. Also, associated with each row and column is a unidirectional ring which passes through all stations in the given row or column. Note that the same network topology has been used in a bridged-ring design [8]. The rings will be identified by their respective row or column IDs. For example, the top row ring is referred to as row-ring 0 and is abbreviated by RR0. Similarly, the rightmost column ring in Fig. 1 is identified as column-ring 3 or CR3. Note that to simplify the description of the network, all of the row ring directions point to the right and the column rings point towards the bottom of the figure. The direction

Manuscript received June 18, 1993; revised March 11, 1994; approved by IEEE/ACM Transactions on Networking Editor R. Jain. This work was supported by the Telecommunications Research Institute of Ontario (TRIO).

The author was with the Distributed Systems Research Department, AT&T Bell Laboratories, Murray Hill, NJ. He is now with the Communications Research Laboratory, the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ont. L8S 4K1, Canada.

IEEE Log Number 9403693.

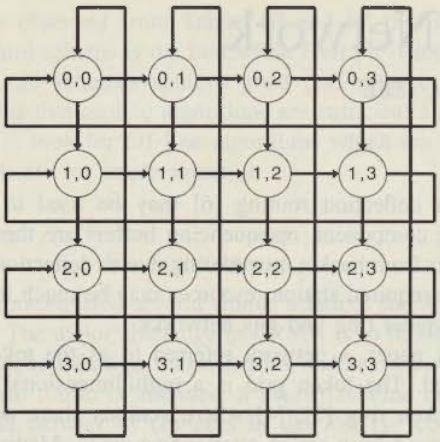

Fig. 1. The token grid (4 × 4 example).

of ring rotation is arbitrary and all subsequent discussion applies for Manhattan Street [6] reference directions also. It should also be noted that arbitrary numbers of stations can be accommodated simply by growing the network one row (or column) at a time. The new ring need not be fully populated. In this expository paper however, we consider only square networks (i.e., $R = C$).

In an implementation of the token grid, Fig. 1 represents an example of the virtual topology. The actual physical topology would more likely consist of a wiring concentrator with station fibres terminating at one or more centralized hubs [2], [4]. At the hub, passive connections are made between the fibres. It can be seen from Fig. 1 that each station in the token grid has both two transmitters and two receivers. This is the same as a dual ring network such as FDDI [4]. Thus in a wiring concentrator layout, the total amount of transmission hardware and fiber is identical for the two networks. Note that as in a typical ring, no transit packet buffering is provided in the stations except for the usual few bits of station latency.

Access to the various rings is determined by tokens which circulate throughout the network. Before an exact mechanism is discussed, we first consider the basic station configurations shown in Fig. 2. Note that the two configurations may be implemented by a pair of 2 × 1 selectors at the station. Fig. 2(a) shows the double-ring (DR) connection. When in this configuration, the rings passing through the station are decoupled and it has access to each one independently. Fig. 3 shows a token grid where all stations are in the DR configuration. In this case, independent communication is possible on all of the row and column rings. To transmit on a given row or column ring, a row-token (RT) or column-token (CT) must be seized in a manner identical to the operation of a conventional token ring [2,4].

When a station is in the single-ring (SR) configuration of Fig. 2(b), the corresponding row and column rings have been concatenated into a single row/column ring. This ring is referred to as either a row/column ring (RCR) or a column/row ring (CRR). When this happens, the two rings are said to have been *merged*. An example is shown in Fig. 4 where station (1,2) has merged row ring 1 and column ring 2 thus forming RCR(1,2) (or CRR(1,2)). When a row/column ring

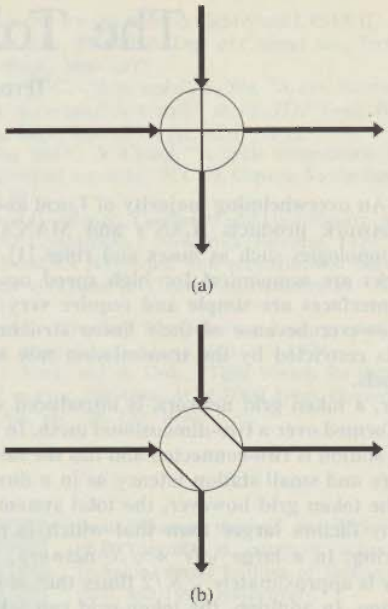

Fig. 2. Station configurations.

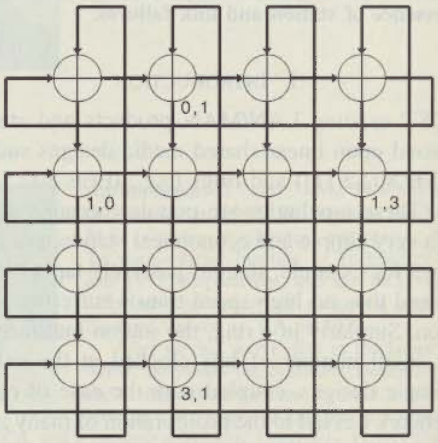

Fig. 3. The token grid (all stations in DR state).

is formed, two special tokens control media access to it. The first is referred to as a row/column token (or RCT) and the second is a column/row token (or CRT). In Fig. 4, the RCT is captured by stations on row 1 that transmit to stations on column 2. Similarly the CRT is used by stations on column 2 to transmit to stations on row 1. A protocol is used to ensure that at any time there is only one token active on any merged or unmerged ring. However note that there may be a number of stations which are simultaneously merged, provided that they do not share any of the same rows or columns. This restriction is enforced by a two-dimensional token passing algorithm to be discussed in the next section. It should also be noted that extra bits are added to identify the tokens as being either RT, CT, RCT or CRT. In the row/column and column/row tokens, the connected column or row is also identified.

III. Protocol Operation

A basic version of the token passing mechanism is now introduced. As discussed above, each row and column ring

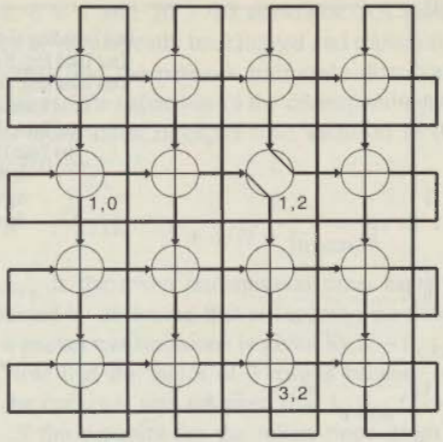

Fig. 4. The token grid (station (1,2) in SR state).

has an associated token. In Fig. 1, for example, station (1,2) could access row ring 1 (RR1) by capturing row-token 1 (RT1) and column ring 2 (CR2) by capturing column-token 2 (CT2). Transmission to a station on a different row and column can only be accomplished when a ring merge has occurred. The merging process is identical for all stations and is now described. A formal description of the basic algorithm is shown in Protocol 1.

Consider a column token j arriving to a particular station (i, j). When this happens, the token is always captured even though the station may have nothing to transmit. In the event that the station does have a packet for column ring j, it is permitted to transmit one packet at this time. The station continues to hold the column token until the row token appears. These actions are shown in the first section of Protocol 1.

When a row token arrives to a station which is not holding a column token, the station is free to capture it and transmit onto the ring in question. Afterwards, the token is released in the usual way.

When a row token i arrives at a station (i, j) which is currently holding a column token, the row token is always captured. At this time, the station may transmit a packet onto row ring i if desired. The station is now holding both row and column tokens and switches to the SR configuration thus merging its two rings. This is shown in the second section of Protocol 1. The station then issues a row/column token onto the merged ring through its row ring output fibre. Stations on this row are now able to capture this token and transmit to stations on the connected column. Note that this mechanism is similar to that used in the bridge design discussed in [7]. Any non-merged station that sees a row/column token is free to capture it and transmit onto the merged ring. Following this, the token is released.

Once the row/column token returns to station (i, j), it is captured. At this time, the merging station is free to transmit a packet onto the merged ring. This action is shown in section 3 of the algorithm description. The token is then converted into a column/row token and circulated onto the station's column output fibre. This allows stations on the column to transmit packets to the connected row. As in the row/column case, any non-merged station that sees a column/row token is free to capture it and transmit. The token is released afterwards.

Eventually the column/row token will return to the merging station. At this time the token is captured and the station may again transmit a packet onto the merged ring. Following this, the station switches to the DR state and releases the original row and column tokens onto their respective rings. This is shown in the last section of the protocol description.

Using the algorithm described above, full network connectivity is achieved and governed by the rotation of the row and column tokens. However, it is apparent that transmissions on a given row or column ring do not necessarily occur in a contiguous fashion, that is, with the ring in an unmerged state for the duration of all transmitted column or row ring packets. Instead, individual transmissions may be interspersed with packets transmitted during ring mergings. In the resulting network, CT's rotate about each column, generating ring merges. Similarly, at each row, CT's are serviced by the row token, by creating merges in a fair and round-robin fashion.

In addition, it can be seen that if a station is permitted to capture a RT whenever it appears, then row-to-row traffic may be preferred over all others since column tokens must wait as part of the merging procedure. This advantage may be easily overcome by either restricting all transmissions to RCT and CRT captures, or by permitting only one RT capture each time a CT is observed. It is clear that this restriction may unnecessarily reduce the total throughput in certain nonuniform load situations. For example, consider the case where n stations on a given column are the only ones active, with packets to be sent on that column only. The above restriction will prevent any of these stations from using the full

PROTOCOL 1:
BASIC TOKEN GRID PROTOCOL

Column Token j Arrives To Station (i, j) :	
1.	Capture the token.
2.	If a packet is available, transmit onto Column Ring j.
3.	Hold Column Token j. Wait for Row Token i.
4.	Done.
Row Token i Arrives To Station (i, j) :	
1.	Capture the token if a packet is available for Row Ring i or if Column Token j is being held.
2.	Transmit onto Row Ring i if a packet is available.
3.	If the station is not holding Column Token j then release Row Token i and go to 5.
4.	Enter the SR state. Generate and release Row/Column Token i, j.
5.	Done.
Row/Column Token i, j Arrives To Station (i, k) :	
1.	Capture the token if a packet is available for Row/Column Ring i, j or if $k = j$.
2.	If a packet is available then transmit onto Row/Column Ring i, j.
3.	If $k \neq j$ then release the token and go to 5.
4.	Generate and release Column/Row Token i, j.
5.	Done.
Column/Row Token i, j Arrives To Station (k, j) :	
1.	Capture the token if a packet is available for Column/Row Ring i, j or if $k = i$.
2.	If a packet is available then transmit onto Column/Row Ring i, j.
3.	If $k \neq i$ then release the token and go to 5.
4.	Enter the DR state. Regenerate and release Column Token j and Row Token i.
5.	Done.

bandwidth of its row ring. If a given station is using only a fraction $1/n$ of the available column ring capacity, then at most it can achieve the same throughput on its row ring. Although the system is fair in this case, the total throughput has been reduced over that which is possible without this restriction.

In the scheme described above, a station passively waits for the appropriate token in order to transmit to a particular destination. However, it is pointless to merge a pair of rings if there are no stations on them wishing to communicate. A more efficient mechanism which also reduces delay under light loading may be used in this case. Each station is equipped with two state variables known as the Column Reservation Variable or CRV and the Row Reservation Variable or RRV. When CRV (or RRV) is set at a station, it indicates that some station on the associated row (or column) ring would like to acquire the column (or row) ring of the station. For example, if station (1,0) in Fig. 1 has a packet queued for station (3,2), it would set the CRV in station (1,2). This information is used by station (1,2) to determine whether a RCT (or CRT) is needed at the next merge opportunity. When a column token arrives at a station, the station checks both CRV and RRV. If either of CRV or RRV are set, then the station holds the token and waits for the arrival of the row token. When the row token arrives, the station generates either a CRT (if CRV=0 and RRV=1) or a RCT (if CRV=1 and RRV=0). Note that if CRV=RRV=1, the station generates a RCT first and then a CRT after the RCT returns.

The mechanism whereby a station sets the CRV (or RRV) bit in another station is very simple. Each row (column) token has a reservation flag bit for each column (row). Whenever a row (column) token passes a station, it checks to see if it has packets queued for any other column (row). If it does and has not yet made a reservation, it sets the corresponding bit(s) for the destination column (row) associated with the packets to be transmitted. When this token arrives at a station, the reservation flag bit for that column (row) is inspected. If the flag is set, then the station sets CRV=1 (RRV=1) and resets the reservation flag in the token. The same reservation flags are also included in the header of each packet. This allows a station to set the CRV (RRV) bit in the station at the beginning of each packet transmission.

It can be seen that transmission activity in the token grid has properties unlike conventional LAN's. As in other mesh networks, contention for various sets of links (i.e., rings) will be different depending upon destination traffic patterns. Naturally, stations contending for more bottlenecked rings will experience higher delays than others. Such a situation is unlike that in LAN's where a single channel must be shared by all transmitted packets, regardless of destination flows. It will be seen that because of the added transmission concurrency, station performance is much better in many common situations.

Throughout this paper, it will be assumed that a station may transmit at most one packet whenever a token is captured. In addition, the early token release mechanism used in FDDI [4] is employed. Thus, whenever a token is to be released following a packet transmission, it is done immediately after the transmission is completed. Versions using delayed token release [2] are also possible, but are much more sensitive to propagation delay [10].

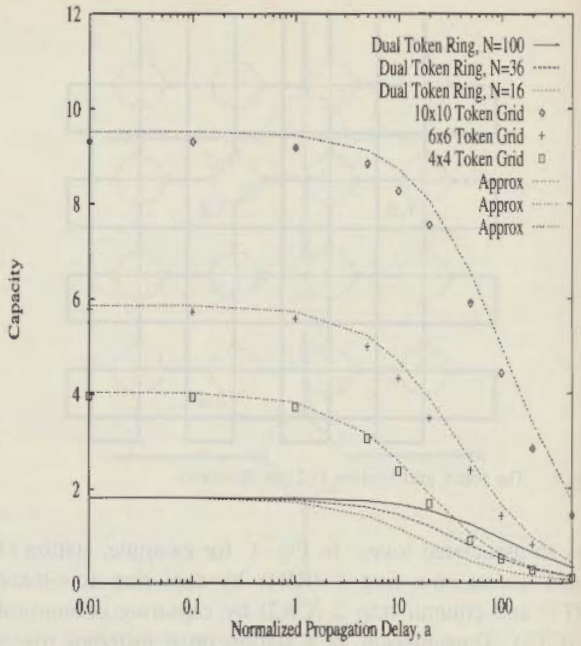

Fig. 5. Uniform load capacity.

IV. Network Performance

In this section, we discuss the performance of the token grid network. As a benchmark, the performance of a dual ring is used for comparison. In a dual ring (such as FDDI [4]), the transmission hardware requirements are identical to that of the token grid. Each station is equipped with two transmitters and two receivers. In addition, each station in both networks may be implemented with a very short station latency. In order to ensure a valid comparison, it is assumed that only one packet may be transmitted per token seizure in both networks. Since the token grid topology allows concurrent transmission of packets on disjoint routes, it is assumed that in the station implementation, separate packet queues are maintained for each column and row. This simple function would typically be implemented by a set of pointers in a low level device driver or VLSI DMA controller. For comparison purposes, we assume a wiring hub layout for both networks. It is assumed that the distance from each station to the wiring center is the same for all stations. The token ring consists of N stations with a total ring latency defined to be τ seconds. The station-to-station latency is thus τ/N seconds. In the results, our attention is restricted to square token grids with $R = C$ and a total of $N = R \cdot C$ stations.

The maximum throughput or capacity performance of the token grid is considered first. Since the token passing mechanism yields complex cycle patterns, exact simulation results for network capacity are first given. Subsequently, a simple capacity approximation will be described.

In Fig. 5, the normalized network capacity is shown plotted against the normalized total ring latency $a = \tau/T$ where T is the packet transmission time in seconds. Results are given

for 4×4, 6×6 and 10×10 networks. All stations were assumed to be permanently backlogged and transmitting fixed-length packets into the network uniformly distributed across all destinations. The capacities of the corresponding 16, 36 and 100 station dual token rings are also included in the graphs and are given by

$$C_{\text{DTR}} = \frac{2}{1 + a/N + t_{\text{token}}/T} \tag{1}$$

where t_{token} is the token transmission time. Equation (1) is easily obtained by observing that at capacity, the time between successive packet transmissions is given by $T + t_{\text{token}} + \tau/N$ seconds. Note that the factor of 2 results because of the two rings. In the curves it was assumed that $t_{\text{token}}/T = 0.1$.

In Fig. 5 the capacity for the token rings drops off from a value slightly less than 2 in the usual fashion. However, the total throughput for the token grid is much higher. In the 16 station example, the capacity is more than 100% higher for small values of a. Also it is important to note that unlike conventional LAN's [4], the total capacity of the token grid is an increasing function of the network size. In the 100 station case, the total throughput is more than 5 times that of the token ring at small values of a. These improvements result from the transmission concurrency obtained by using a network topology which supports multiple concurrent transmissions.

Under the conditions shown, a simple and accurate approximation is given for the uniform traffic capacity. The approximation is obtained by assuming that network activity is cyclic and that the column tokens are never idle. To generate this behaviour, assume that the packet lengths are fixed and that all row/column transmission modes are exactly synchronized. In Fig. 1 for example, assume that at some time instant stations (3,0), (2,1), (1,2) and (0,3) have all created row/column tokens. Since all stations are busy, transmission activities and timing on the 4 row/column rings will be identical and a series of simultaneous transmissions occur. After the RCT and CRT rounds are completed, each row and column token are also used by the following stations at the same time. This results in 8 concurrent row and column ring transmissions followed by a new set of four ring mergings. We denote a cycle to be the time between these merging instants. It can be seen that in this example, four cycles identical in duration occur before the same set of stations merge again. Employing a usual cycle analysis, the capacity can be shown to be

$$C_{TG} \approx \frac{\sqrt{N} + 1}{(1 + t_{\text{token}}/T)(1 + \frac{1}{2\sqrt{N}}) + \frac{a}{2N}(2 + 1/\sqrt{N})}, \tag{2}$$

or

$$C_{TG} \approx \frac{\sqrt{N}(\sqrt{N} + 1)}{2\sqrt{N} + 1} C_{\text{DTR}} \tag{3}$$

In deriving the above equation, the total normalized throughput per cycle is given by $U = R \cdot 2(R + 1)T$. In this cycle, R stations simultaneously release row/column tokens. For each of these concurrent cycles, there are a total of $2R+2$ packets transmitted. One is associated with each of the row and column tokens, and the other 2R are for the stations which transmit using the RCT and CRT tokens. This gives the value for U. To obtain the final result, U is divided by the average time per cycle L. This is easily shown to be $L = T + 2R(T + \tau/R^2 + t_{\text{token}}) + t_{\text{token}} + \tau/R^2$. In Fig. 5, results are shown for this capacity approximation. As expected, it tends to slightly overestimate the true capacity since in a real system, column tokens would have a nonzero waiting time. It can be seen however that the approximation is a very good one. Also, it can be seen that when N is large, $C_{TG} \approx \frac{\sqrt{N}}{2} C_{\text{DTR}}$.

It is also interesting to consider a comparison of the token grid to a dual ring under light load conditions. We will assume that the merge reservation mechanism discussed in Section III is used and that tokens are received entirely before being relayed. In this case the worst-case delay for a square token grid is easily shown to be $(5R - 2)(t_{\text{token}} + \tau/N) + T + (2R - 1)\tau/N$. The corresponding expression for a dual ring is $R^2(t_{\text{token}} + \tau/N) + T + R^2\tau/N$. Thus the light load delay is proportional to N for a dual ring and to $\sqrt{N}$ for the token grid. A similar analysis for mean delay under light load results in the same conclusion. The above result shows that when $R > 4$, the token grid delay is always better. When this is not the case, the dual ring delay may be better depending upon the relative magnitudes of t_{token} and τ.

A. Mean Delay Bound

In this section, a lower bound is derived for the mean waiting time of packets under uniform symmetric traffic conditions. In the token grid protocol, it is readily apparent that there are many variations in the how stations transmit packets to a given destination. For example, station (0,0) can transmit to station (0,3) upon capturing either row token 0 or any of the RC tokens generated by stations on row 0. To simplify the analysis, we will assume that all traffic in the network is transmitted using only RC and CR token captures. Stations thus transmit to others on the same row or column after capturing the appropriate RC and CR tokens generated by other stations on the same row or column. Accordingly, each station maintains a set of $2(R - 1)$ queues, corresponding to the identities of the RC and CR tokens it captures for transmission.

Consider a cycle which consists of the total excursion time of a particular column token around its column. For simplicity, we will suppress all time indices and view the rotation of the token in cycle 0. Also for convenience we will consider the column 0 token ($CT0$) starting its cycle at station (0,0). Note that this cycle comprises R subcycles each consisting of the time spent at each of the R rows before returning to row 0. At any particular row r, the time delay incurred by the column token is given by the sum of three terms, namely

$$W_{RT}^r + T_{RC}^r + T_{CR}^r \tag{4}$$

where $r \in \{0, \ldots, R - 1\}$ where

W_{RT}^r = waiting time for the row token r to arrive at station $(r,0)$

T_{RC}^r = total duration of the RC token rotation cycle at row r

T_{CR}^r = total duration of the CR token rotation cycle at row r

The two token rotation cycles are associated with the merging of station $(r, 0)$ once the row token arrives. Note that

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.