Understanding Encrypted Networks Through Signal and Systems Analysis of Traffic Timing

David Cousins, Craig Partridge, Kevin Bongiovanni, Alden W. Jackson, Rajesh Krishnan, Tushar Saxena, and W. Timothy Strayer
BBN Technologies 10 Moulton Street, Cambridge, MA 02138
{dcousins, craig, kbongiov, awjacks, krash, tsaxena, tstrayer}@bbn.com

Abstract—Recent studies have shown that signal-processing techniques are quite valuable for the modeling and analysis of modern networks and network traffic [1] [2]. However, to date most of these studies have focused on characterizing the multi-scale and long-memory stochastic nature of single streams or traces of non-encrypted network traffic. The key approach used has been to transform traces of packet arrival times and/or packet size into encoded time signals, which then allow analysts to perform standard statistical and time-frequency-scale signal analyses. In this paper we summarize some of our results which show that under this analysis, traces from both wireless and wire-line networks leak useful information about the properties of the network and applications under examination, even when the actual packets are encrypted or attempts are made to mask the traffic timing. Furthermore, when multiple signal techniques are used between individual time streams, even more information about the underlying routing and flows can be uncovered.

1. Introduction

In this paper we summarize the results of several experiments where we have performed traffic analysis on simulated data networks using only minimal information about packet transmission timing and elementary signal and system analysis techniques. Each analysis targets a particular layer of the network and its applications. The algorithms used for each layer will be briefly summarized, but where appropriate the reader will be referred to previously published details.

In particular the results presented will show:

- Standard spectral analysis can extract timing information, such as round-trip times of TCP connections, from traces of aggregated data. Such information can help identify the type of transport protocol being used, and can help isolate the relative locations of the connections within the network.
- Cross-spectra analysis of the traces from nodes in a network allows us to perform network topology discovery (i.e. determine how packets are routed through the network), even with moving nodes and dynamic routing.
- State-space analysis of the traces can be used to identify network topology as well as determining which nodes of a wireless network provide routing ingress and egress with other distant, non-detectable nodes.
- Graph-Theoretic based routing algorithms can further process the above topological information to develop estimates of the most likely paths for distinguishable traffic flows.

2. Network Understanding Techniques

Discovering Transport and Application Layer Timing using Spectral Analysis

Simple spectral analysis can be used as a tool for network understanding at higher layers, such as the transport and application layers. These layers are typically concerned with establishing end-to-end communication sessions between end-hosts. This includes network transport protocols such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), up to applications such as video streaming, IP telephony, FTP (File Transfer Protocol) and HTTP (Hyper Text Transfer Protocol or web browsing). We are concerned with identifying key timing characteristics. For example, timing properties such as round trip times, and application data-transfer characteristics such as packet send-rates.

A Typical Wireless Network Analysis—In wireless networks, we use a *network tap* model which consists of a system that can: 1) detect physical layer RF transmissions above a certain threshold, and 2) uniquely associate each transmission received with a unique node identifier [3]. Consequently, a tap may only hear a subset of nodes in the network. Moreover, we do not assume that the taps participate (or, indeed, even know about the Media Access or MAC layer) in the network. We have applied the same technique to wireline networks with similar results, using tools like *tcpdump* and data filtering based upon the unencrypted packet headers.

We now present the results from [3], which analyzed a four node ad hoc wireless network with two *Constant Bit Rate* (CBR) flows and one *File Transfer Protocol* (FTP) flow present, as shown in Figure 1. We simulated this network using the well known *ns-2* simulator [4], using 2Mb/s transmission bandwidth (the *ns-2* settings for Lucent WaveLAN), an 802.11b MAC layer [5], and the Dynamic Source Routing (DSR) protocol [6] to maintain connectivity in the ad hoc network. We generated an analysis signal for 300 seconds of network activity using a simple encoding scheme: the signal

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract No. MDA972-01-C-0080. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied.
0-7803-7651-X/03/$17.00 © 2003 IEEE
IEEEAC paper # 1101

is created from the tap trace by assigning a $+1$ at the time of all transmissions from node 0, and -1 at the time of all transmissions from node 3. As an aside, the trace produced by this tap is more complicated than a trace from a wire-line network with an equivalent topology. This is due to the support traffic used in wireless networks: DSR routing updates which do not correspond to any end-to-end flow; 802.11b RTS, CTS and MAC layer ACK transmissions; and collisions resolved at the MAC layer that lead to retransmissions. Furthermore, the wireless trace may be noisy or incomplete (due to missed or false detections arising from variable received SNR at the tap).

This analysis signal is an example of a *non-uniformly* sampled signal. Lomb, Scargle, Barning, Vaníček [7][8] developed a spectral analysis technique specifically designed for such kinds of data. The method estimates a *power spectrum* for N arbitrary time points of data at any set of arbitrary frequencies $f_1, \dots f_M$. The result is equivalent to a least-squares fit of a sinusoid of frequency f_i to the data points. Figure 2 shows the resulting periodogram on the analysis signal, plotting power vs., period (i.e. $1/f_i$), which is more useful than frequency for identifying network timings. The three prominent peaks correspond to key periodicities in each of the three flows. Both CBR flows are revealed by peaks almost exactly at their transmission rates. The FTP flow from $0 \rightarrow 3$ can be identified by the peaks spread around 328.85 ms, which corresponds to the average round-trip time (RTT) for this flow, and falling well within the RTT standard deviation of 92.5 ms (both numbers reported by *ns-2*).

In this example, the method was able to identify the key timing parameters, revealing all three IP flows, even though the tap signal did not contain any transmissions caused by one of the flows. Observe that the plot clearly shows the period of CBR $1 \rightarrow 3$ even though the tap cannot hear the transmissions of either node 1 or 2, and has no way to know when node 3 receives a packet. We speculate that this arises from one flow imprinting on the other. The CBR flow from $1 \rightarrow 3$ shares part of its path with the FTP, causing timing interactions between the two flows. The CBR timing is thus reflected in the timing of the FTP acknowledgements. Work with other traces, real and simulated, have so far supported this speculation.

Identifying Active Links and Network Dynamism via Cross-Coherence

The next technique presented performs joint analysis of multiple trace signals in order to relate transmissions in one location with those at another. We have found that standard windowed time-frequency techniques such as spectrograms made with *cross spectral densities* (CSDs) or their normalized counterpart, *cross-coherence*, capture the variations in these signal relationships as they evolve over time, and can reveal the dynamic changes in data flows due to dynamic network routing, or changes in network loading. If there is enough periodicity in a trace to show spectral peaks, and if the transmissions of one source are forwarded or answered by another source at some layer of the network (e.g. TCP ACKs or MAC RTS/CTS protocols), then we can compute the degree that the two different traces are related using CSDs or cross-coherence.

We now show an example, again from [3], where we have identified all active single-hop or MAC layer links between the various nodes in a *mobile ad hoc wireless network*. Such networks have transient links as the node connectivity changes. The changes are due to nodes moving in and out of range, and passing behind blocking obstacles.

The trace data is generated from the same simulated wireless network described in Figure 1, except with a reduced set of flows: an FTP from $0 \rightarrow 3$ by way of node 2, and a CBR from $1 \rightarrow 3$, also by way of node 2. We generated a set of analysis signals for 30 seconds of network activity using a multiple signal encoding scheme: a separate, uniformly sampled time signal for each node is generated by placing a unit impulse at the start time of each detected transmission (quantized the the nearest sample).

Network re-routing was triggered twice during this 30 second period by having node 1 move around node 2 at a constant speed, stopping briefly between nodes 0 and 2, then between nodes 2 and 3, finally returning to its original location. When the distance between 1 and 3 becomes smaller than that between 2 and 3, the network re-routes the $1 \rightarrow 3$ traffic directly between those two nodes.

Cross-coherence spectra of 32 frequency bins computed for each node pair were computed using 512 ms time windows and a 50 % overlap. The resulting two-dimensional time-frequency grams for each pair of nodes are displayed in Figure 3. Plots with steady spectral peaks (i.e. horizontal lines) indicate strong, continuous coherence, suggesting steady two-way transactions (and hence a flow caused by an application). Furthermore, as mentioned in the previous section, the location and shapes of the peaks provide information that may help identify the types of data transfers.

Initially, strong coherence occurs between node pair 2 and 3 The link $2 \leftrightarrow 3$ is carrying the both the FTP, and CBR. There is a lack of coherence between nodes 0 and 1 because they do not share any flows.

When the network reroutes, most cross-coherences change quite abruptly, with the strongest changes in the plots for $1 \leftrightarrow 3$ and $2 \leftrightarrow 3$. Thus we can detect changes in the network flows due to re-routing. The detection of such changes are easily automated, and we have used this technique to generate dynamic topology maps with a reasonable degree of accuracy.

Link Topology and Source-Sink Discovery Using State-Space Analysis

A State-Space Technique for Generating a Conversation Probability Matrix—This techniques examines the structure of all recorded traces from a purely causal perspective. The underlying approach is based on fundamental assumptions about the recorded traffic flows from the perspective of a given event (transmission), rather than from the perspective of the network node.

The algorithm is based upon two assumptions. 1) In general, the likelihood of a transmission being a response to a *prior* transmission decreases as the elapsed time between them increases. This is a direct consequence of causality with the added stipulation that a network tries to operate efficiently. Loosely speaking, we expect related packets to be located (temporally) closer than unrelated ones. 2) The inter-arrival times between a *fixed* event and any other event are approximately Poisson distributed [1]. This assumption may seem somewhat counter intuitive in light of recent research in the long range dependency (related to self-similarity or fractal structure) of network traffic as observed in [1]. However, this reference provides a metric for this dependency H, the Hurst Parameter. We have empirically measured H for all our simulations, and have found that long range-dependency behavior can be neglected for time scales on the order of 4.5 ms or less. The average inter-event times for our networks occur on the order of 1 ms, which falls comfortably within the time scale where traces can be characterized by classical second-order statistics such as the Poisson distribution.

The algorithm uses a state-space representation of the nodes to generate a *conversation probability matrix* (CPM). The state vector of node j is updated at each detected transmission by node j by computing a weighted sum using the previous state vector and a new weight vector W_j consisting of weighted contributions from each node i. Each resulting i, j is an entry in the CPM and corresponds to the probability that the transmission generated by node j is due to one generated previously by node i. Figure 4 shows a notional representation of the weighting assignment. The constant λ is the Poisson parameter estimated from the trace statistics. One further aspect of the algorithm results from the causality constraint: the weight for node i is set to zero if an instigating transmission from node i ended after the start of the current transmission from node j.

Identifying Network Sources and Sinks—This technique has been extended to determine the likelihood of a given node being a source or sink for packet flows (either as an application endpoint, serving as a gateway for the wireless network, or as a route to other nodes in the network that are not detected by our tap). We identify such activity as *egress* for simplicity. The basic assumption is that the longer it takes for any *detected* reactions to a transmission to occur, the more likely the current transmission is a response to an undetected node or was simply generated by the node itself. We now define an *egress* weight, corresponding to $1 - \max[W_j]$ where W_j is weight vector generated by the state space technique above. We augment the state j (and hence add a column to the CPM) to include this state of egress. A simple energy or threshold detection over this egress column provides adequate egress node detection.

Link Identification and Egress Identification Performance—This algorithm is run using a sliding window technique to account for network dynamics. We have found that 10 seconds of data is required for the CPM to converge adequately. This process is repeated by sliding the data window forward in time by one second for each batch. The result is a converged CPM, updated once a second.

We have run this algorithm on various simulated wireless networks ranging from 4 to 50 nodes. The algorithm correctly detects 65 to 100% of the links in these networks, including those cases without MAC-layer feedback, and significant volume of broadcast or egress transmissions. The false alarm rate is also low in all cases that do use MAC-layer feedback.

For cases where the entire network can be monitored, the false alarm rate is in the range of 0 to 4.6%. The highest false alarm rate among all cases using MAC-layer feedback is 27% which means that on average only 1.6 false links were detected in each batch. A random guessing strategy will have an expected false alarm rate of 50% in this case.

We have found that in scenarios where all MAC-layer feedback is suppressed, the false alarm rate is quite high. We believe that this is due to the fact that events corresponding to multi-hop forwarding activity, which are hard to differentiate from each other, start to dominate. This is an area of continued research for us.

We finally observe that our algorithm correctly detects 66.7 to 100% of egress nodes for partially observed networks. The false alarm rate varies widely, being as high as 86.9% in one case. We observe that the false alarm rate is higher in cases with complete network observation. This would be expected as there are few or no egress transmissions

Estimating Routes for Wireless Traffic Flows

Moving up the network stack, we present an algorithm for understanding at the network layer (i.e. layer 3). Packets at this layer typically take multiple hops through the network before reaching their destination (an example of this is the IP layer on the Internet). In the domain of ad-hoc wireless networks, we use our algorithm to discover the most likely end-to-end routes that are in use. We define a route between a source-destination pair of nodes as the list of nodes that packets originating at the source and travelling to the destination are known to have taken, or will take, based on the routing tables of the node routers. More than one route may exist be-

[1] We are currently working on extensions to this algorithm that use emperically derived statistical models

tween node pairs. However, we have found that with DSR in our wireless scenarios, there is usually only one dominant route (in terms of traffic volume), even though there may be two or three routes existent between a pair of nodes,

We use the output of the State-Space Technique described in the last section as the input for this algorithm. The input is a trace which identifies the time of each detected transmission, the transmitter ID and the most likely receiver ID. The output is a table of active end-to-end routes. One way of interpreting this output is as an a series of time snap-shot estimates of the routing tables on all nodes in the network.

The route discovery algorithm uses a modified version of the *Floyd Warshall All Pairs Shortest Path Algorithm* (FW-APSP) [9] which produces *all* shortest and equal length routes for each node-pair. We then use an *Aggregate Route Coherence* statistic to break ties when multiple links exist. The algorithm is run over a series of sliding time-windows, allowing it to respond to dynamics in the network.

For each time window, the algorithm performs the following operations. First: Construct a network graph from the input, by creating a link between two nodes (n_1 and n_2) if there are any transmissions from n_1 to n_2. Second: Run the modified FW-APSP algorithm on this graph, with unit weight on each link. Third: Given the resulting routes between each node pair in the graph, if there are multiple shortest routes, find the *most-likely* route for each pair by computing the Aggregate Route Coherence statistic for each candidate route, choosing the one with the largest Aggregate Route Coherence.

The Aggregate Route Coherence is a metric we have developed that indicates the likelihood that packets actually travelled along a given route. It enables comparison of two different equal length routes between the same source-destination pair. This metric is the geometric mean of all the total coherences[2] of each hop of the route in both directions. This geometric 2-way mean has an added advantage of being defined for single link routes.

Route discovery performance— We use a highly stringent scoring for determining performance: a discovered route is correct if and only if all hops in the route are identified, and in the correct sequence. The route discovery algorithm driven by perfect receiver identification (obtained from ground-truth) performs very well. All hops are correctly determined for 79–100% of active routes in the network. The performance is lowest in mesh-like scenarios that offer the greatest number of route choices as well as the longest minimum length routes within the network (where we believe that our scoring may be *too* stringent), and in scenarios with considerable mobility and network dynamics.

The algorithm is highly sensitive to errors in receiver identification, especially if the network has long routes. Intuitively, the number of candidate routes (and the potential for false alarm) increases combinatorially with increase in the false alarm rate for link detection. In a mesh-like scenario, when the receiver identification accuracy drops from 100% (ground-truth) to 78%, the route accuracy drops from 89.5% to 35.6%.

3. Conclusions

In conclusion, we have shown several algorithms that uncover information about the various layers in wireless and wireline networks using only the most minimal of information: lists of transmission times and possibly durations associated nodes in the network. This allows us to examine and understand certain kinds of networks without knowing the contents of the transmissions themselves. These algorithms have limitations, but we have found them to perform very well when monitoring conventional 802.11b networks. The performance of the algorithms do not rely solely on TCP level acknowledgements to detect traffic patterns, as they are sensitive to the underlying MAC layer protocols. However, their performance generally degrades if these MAC layer feedback protocols are turned off.

For Wireline networks, we find the algorithms to be quite robust for understanding UDP and TCP based flows. To date we have limited results with wireline, but feel that there is fruitful research to be done in this area because 1) there is more information available to a network tap (such as source and destination IDs) and 2) there is perfect detection of packet timing as it is picked up by the tap.

There are many potential applications of this technology in, for example, the areas of Network Performance Analysis, Network Intrusion Detection and Monitoring, and as a part of future high bandwidth wireless network and communication systems where the in-situ RF spectra is dynamically analyzed and reallocated instead the static frequency allocations in use today.

4. Acknowledgements

The authors thank fellow members of our staff: Steve Kent, Greg Troxel, Chip Elliott, Alex Snoeren, Keith Carlson, as well as our sponsors Paul Kolodzy, Bob Bonneau and Preston Marshall for their suggestions, directions, and reviews of earlier drafts.

References

[1] O. Cappe, E. Moulines, J-C. Pesquet and A. Petropulu and X. Yang, “Long-Range Dependence and Heavy-Tail Modeling for Teletraffic Data”, *IEEE Signal Processing Magazine*, 2002 V 19 n 3, pp 14-27

[2] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi and D. Veitch “Multiscale Nature of Network Traffic”, *IEEE Signal Processing Magazine*, 2002 V 19 n 3, pp 28-46

[2] Computed as the integral under the cross-coherence vs frequency graphs of section 2.

[3] C. Partridge et. al., “Using Signal Processing to Analyze Wireless Data Traffic”, ACM Workshop on Wireless Security (WiSe), MobiCom 2002, Atlanta Georgia, U.S.A, September 28, 2002.

[4] S. McCanne and S. Floyd, *ns* Network Simulator, `http://www.isi.edu/nsnam/ns/`

[5] *IEEE Std 802.11b — Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE, 1999

[6] D. Johnson, D. A. Maltz, ”Dynamic Source Routing in Ad Hoc Wireless Networks”, in *Mobile Computing*, V353, Kluwer Academic Publishers, 1996

[7] N. R. Lomb, “Least-squares frequency analysis of unequally spaced data”, *Astrophysics and Space Science* V 39, 1976, pp 447-462

[8] W. Press, et al., *Numerical Recipes in C*, 2nd ed, Cambridge Univ. Press, 1995

[9] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, 2nd ed, The MIT Press, Cambridge, Massachusetts, 1990

David Cousins *is a Division Scientist in the High Performance Computing Department at BBN Technologies. He received both the B.S.E.E., and M.S.E.E. degrees from Columbia University. His research interests are acoustics, radar and sonar, novel application of signal processing, real-time hardware in the loop simulation, and embedded high performance computing. He is a senior member of IEEE and OES and is a member of the technical committee for the High Performance Embedded Computing Workshop.*

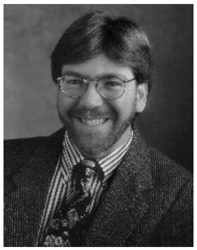
Dr. Craig Partridge *is Chief Scientist for Internetworking at BBN Techologies. He received the A.B., M.Sc. and Ph.D. degrees from Harvard University. His research intersts are high-performance internetworking and traffic security and analysis. He is an IEEE fellow member, and is chair of ACM SIGCOMM.*

Dr. Kevin P. Bongiovanni is a Scientist in the High Performance Computing Department at BBN Technologies. He received the B.S., M.S. and Ph.D. degrees in Mathematics from Rensselaer Polytechnic Institute. His research interests are mathematical applications to industrial problems. He is a member of Sigma Xi, SIAM and an adjunct professor of mathematics at Salve Regina University.

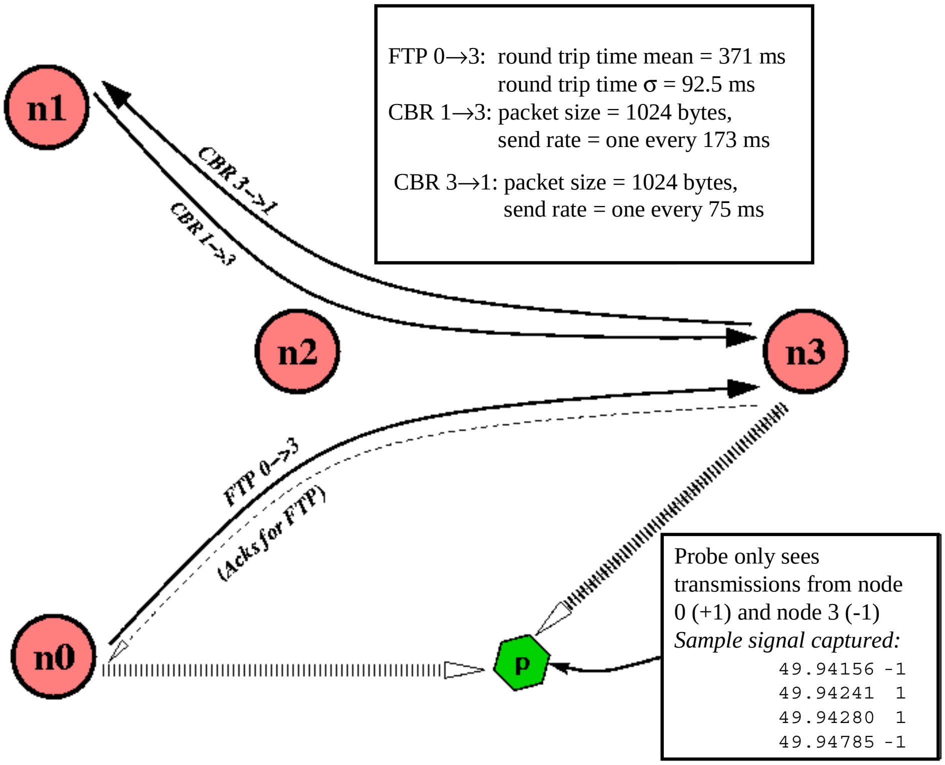
Dr. Alden W. Jackson is a Senior Scientist in the Internetworking Research Department at BBN Technologies. He received the B.A. degree in Physics from the University of Dallas, the M.S.E.E. from Howard University, and a Ph.D. degree in Electrical Engineering from the University of Delaware. His research interests include network architecture, active networks, network management and network security. He is a senior member of the IEEE and ACM, and a member of the technical editorial board for IEEE Network Magazine.

Rajesh Krishnan *is a Senior Network Scientist in the Internetworking Research Department at BBN Technologies. He received the B.E.E.E. (Honours) degreee from the Regional Engineering College, Durgapur, India and the M.S. degree in Computer Engineering from Boston University. His current research interests include distributed algorithms for network self-organization, transport layer protocol performance in satellite and wireless networks, and network security. He is a member of the ACM, the IEEE, and the IEEE Communications Society.*

Dr. Tushar Saxena *is a Senior Scientist in the Internetworking Research Department at BBN Technologies. His research interests include network architecture and traffic analysis.*



Dr. W. Timothy Strayer *is a Division Scientist in the Internetworking Research Department. He received the B.S. degree in mathematics and computer science from Emory University, and the M.S. and Ph.D. degrees in computer science from the University of Virginia. His research interests include transport protocols, active networks, satellite packet switching, virtual private networks, and routing systems. He is an author of* Virtual Private Networks: Technologies and Solutions *(Addison-Wesley) and a senior member of the IEEE and ACM*



FTP 0→3: round trip time mean = 371 ms
round trip time σ = 92.5 ms
CBR 1→3: packet size = 1024 bytes,
send rate = one every 173 ms
CBR 3→1: packet size = 1024 bytes,
send rate = one every 75 ms
n1
n2
n3
n0
p
CBR 3->1
CBR 1->3
FTP 0->3
(Acks for FTP)
Probe only sees transmissions from node 0 (+1) and node 3 (-1)
Sample signal captured:
49.94156 -1
49.94241 1
49.94280 1
49.94785 -1

Figure 1. **A Wireless Network With one FTP flow and two CBR flows.** The network is configured to route traffic from nodes 0 and 1 to node 3 (and vice versa) via node 2. The tap is placed such that it only hears transmissions from nodes 0 and 3, and creates a simple signal encoding.

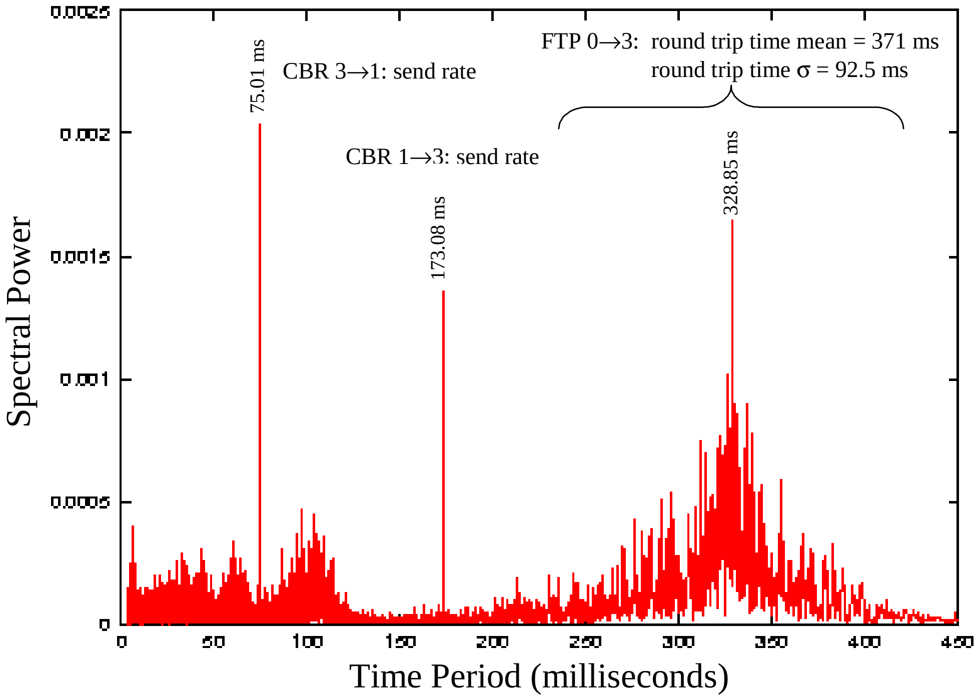


Figure 2. **The Lomb periodogram for the Wireless Network of Figure 1.** The periodogram reveals all three flows in the network, even though the tap can only hear nodes 0 and 3.

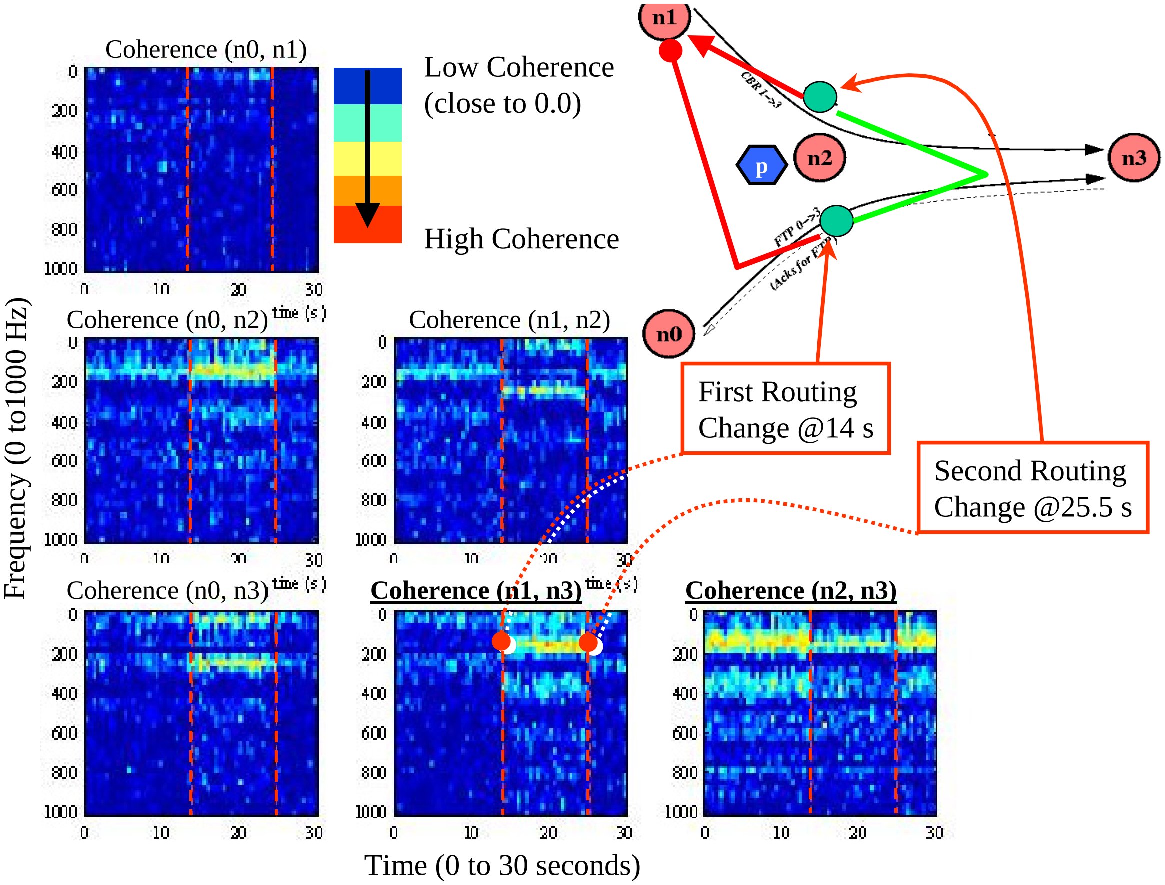


Figure 3. **Coheregrams Showing Time Varying Cross-Coherence Between Nodes in a Mobile Wireless Network.** Re-routing is due to mobile node 1. Coherence changes due to Link/Routing changes are observed at 14 seconds and 25.5 seconds.

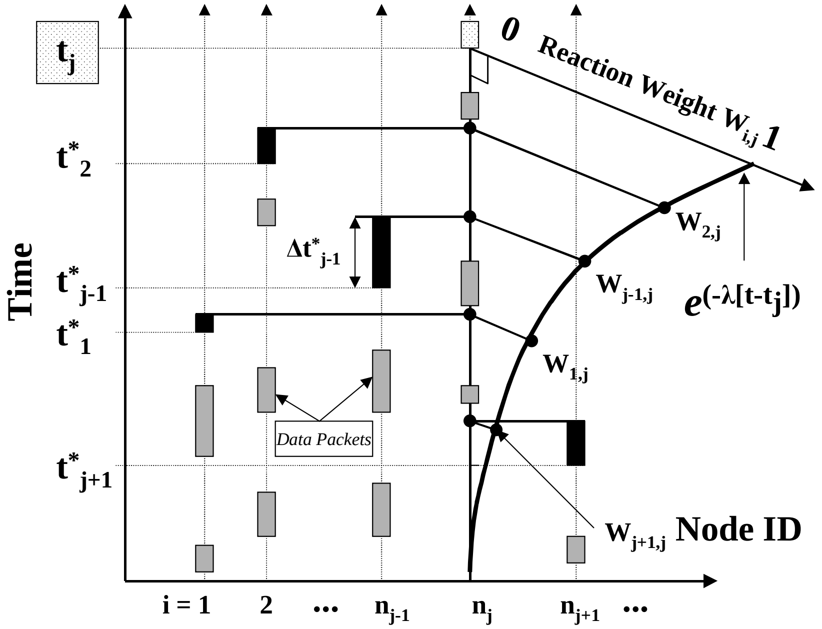


Figure 4. **Notional Representation of the State-Space Weight Assignment Process.** Event t_j has just occured at node n_j (uppermost box on time axis). Each vertical trace depicts packet transmissions from a node (gray with the most recent in black). Every other node is given weight $W_{i,j}$ in vector W_j based upon its value on the exponential tail falling off from t_j.