IP Covert Timing Channels: Design and Detection

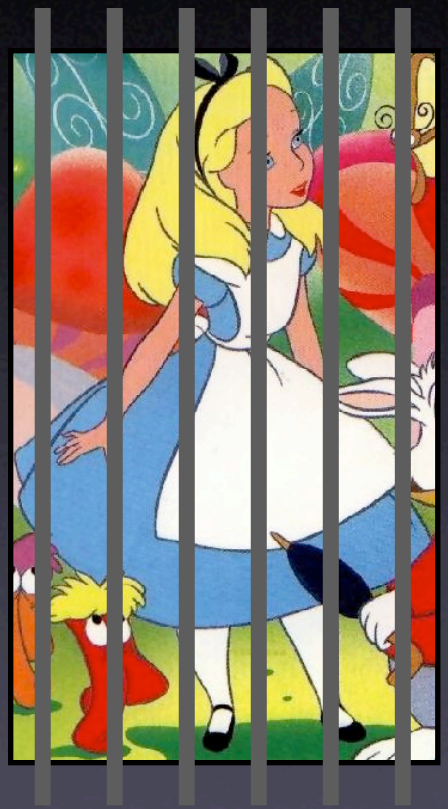
by S. Cabuk, C.Brodley, C.Shields

Presented by
Sam Small
in
Advanced Topics in Network Security (600/650.624)

Outline

- What are covert channels?
- Information hiding and subliminal channels
- Covert channel taxonomy
- The CBS channel
- Evaluating covert channels
- Covert channel detection

The Prisoners' Problem



Alice

Warden

Bob

- Alice and Bob may communicate
- Warden watches all communications
  - If he suspects secret communication or any plans to escape, Alice and Bob will be placed in solitary confinement

Covert Channels

- A means of communication between two processes that are not permitted to communicate, but do so anyway
  - usually done a few bits at a time
  - usually exploits a shared resource or medium
  - traditionally classified as storage-based or timing-based

Most Important

- Covert channels require *COVER*
- Cover should consist of permitted actions and appear innocuous (as not to cause suspicion)
  - We'll talk more about this later

Storage	Timing
Information conveyed by writing or abstaining from writing	Information conveyed by the timing of events
Clock not needed	Receiver needs clock

The Disk-arm Channel

- A covert channel involving the placement of the arm of a shared disk-drive in KVM-370 (1979)
- Involves a shared disk drive with adjacent cylinders shared for read-access by two virtual machines
  1. HIGH operates with high secrecy level
  2. LOW operates with low secrecy level

- The channel is timing-based
  - HIGH and LOW issues a series of read requests to the disk
  - By measuring the seek time of these requests, HIGH can leak data to LOW

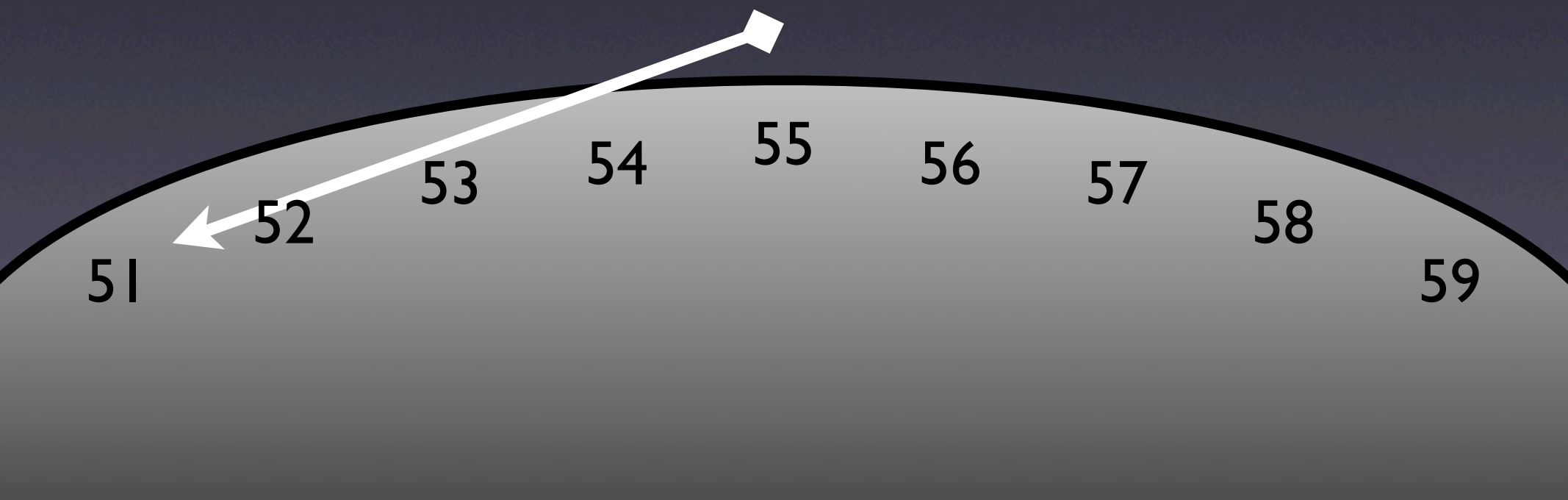
An Analysis of Covert Timing Channels (1991)

- Presents a variant that requires no external clock
- Variant channel characterized through the order of events
- Suggests new model: *storage-nature* and *timing-nature*
  - The usefulness of this distinction is questionable

The Disk-arm Variant

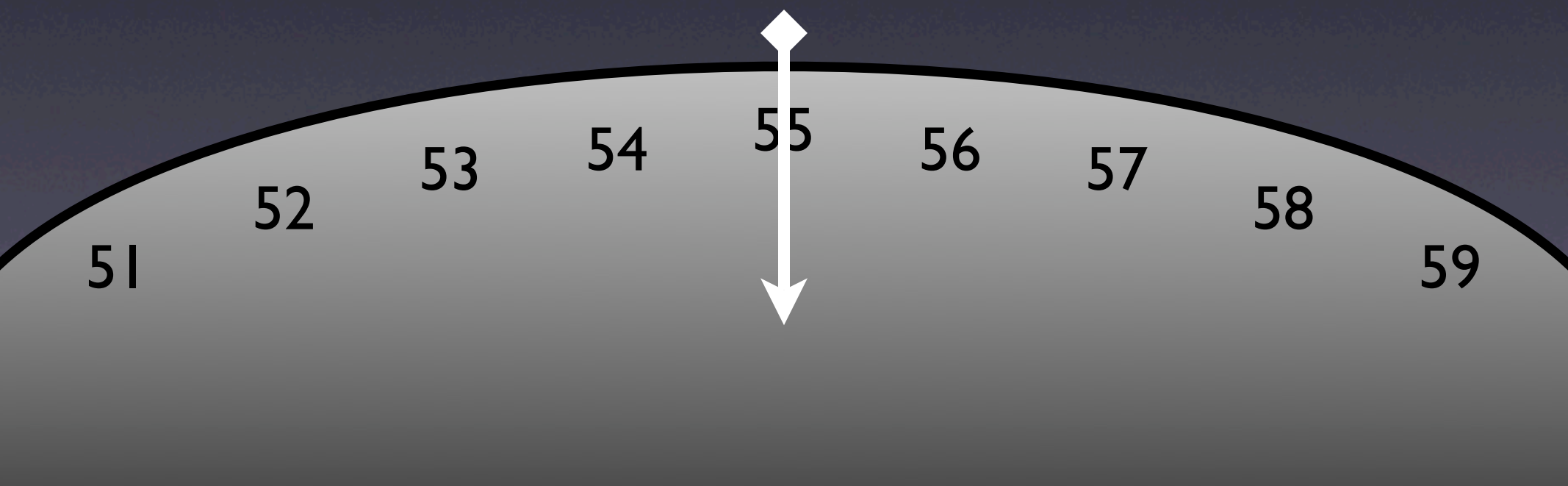
LOW HIGH

Read @ 55



The Disk-arm Variant

LOW HIGH



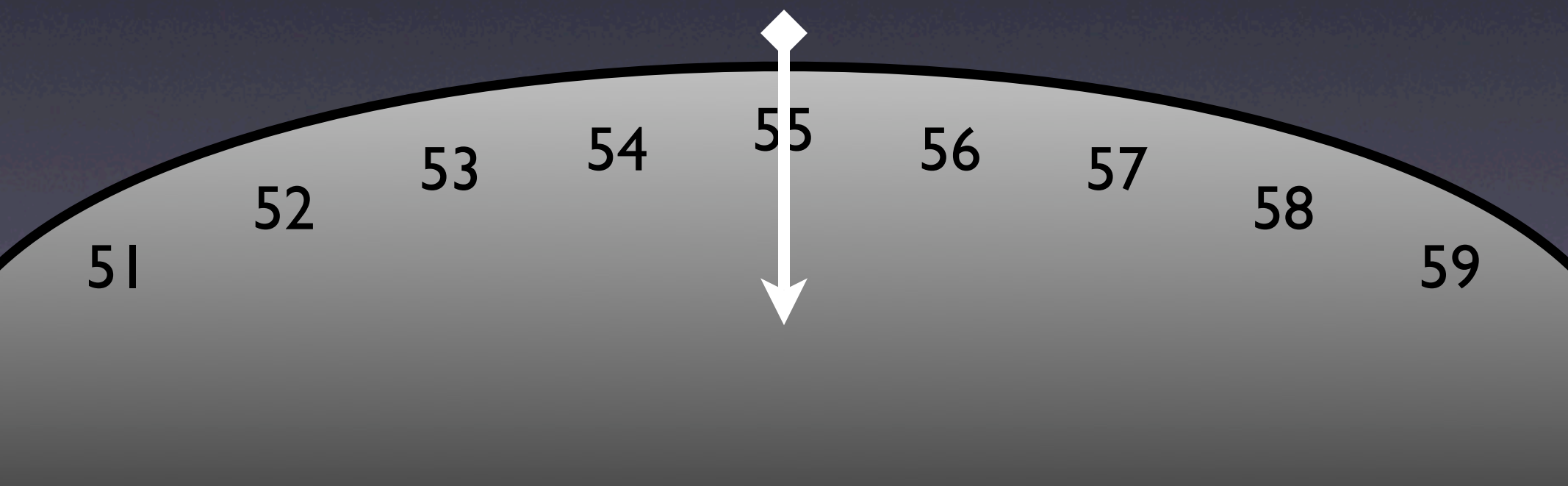
The Disk-arm Variant

LOW

HIGH

to send 0: Read @ 53

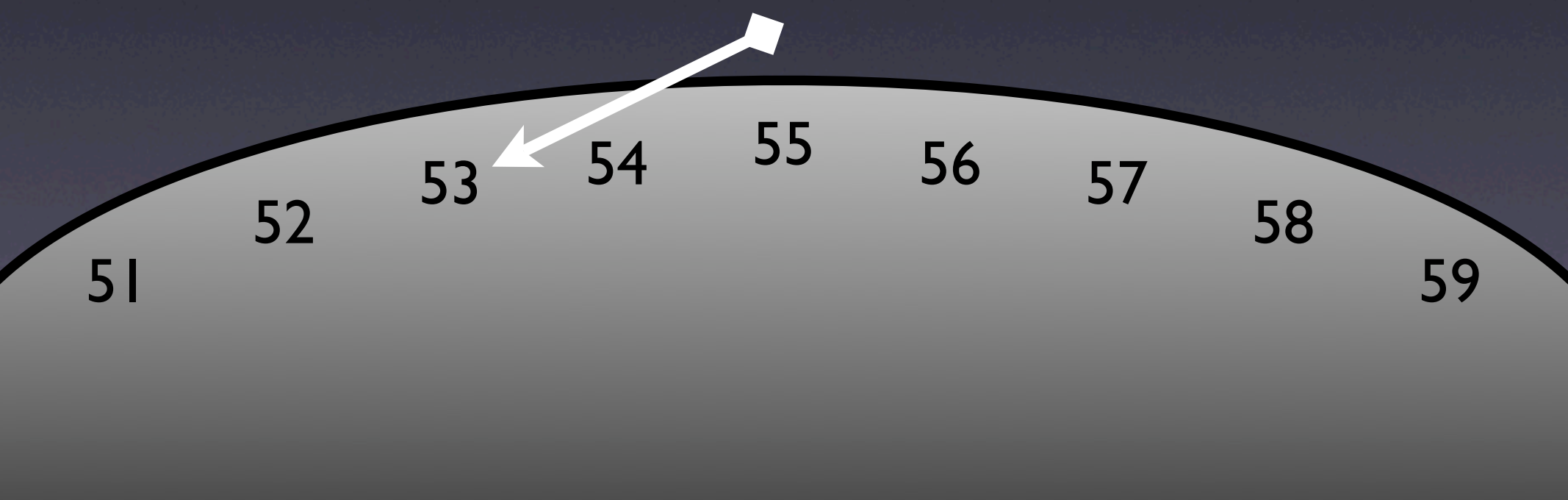
to send 1: Read @ 57



The Disk-arm Variant

LOW HIGH

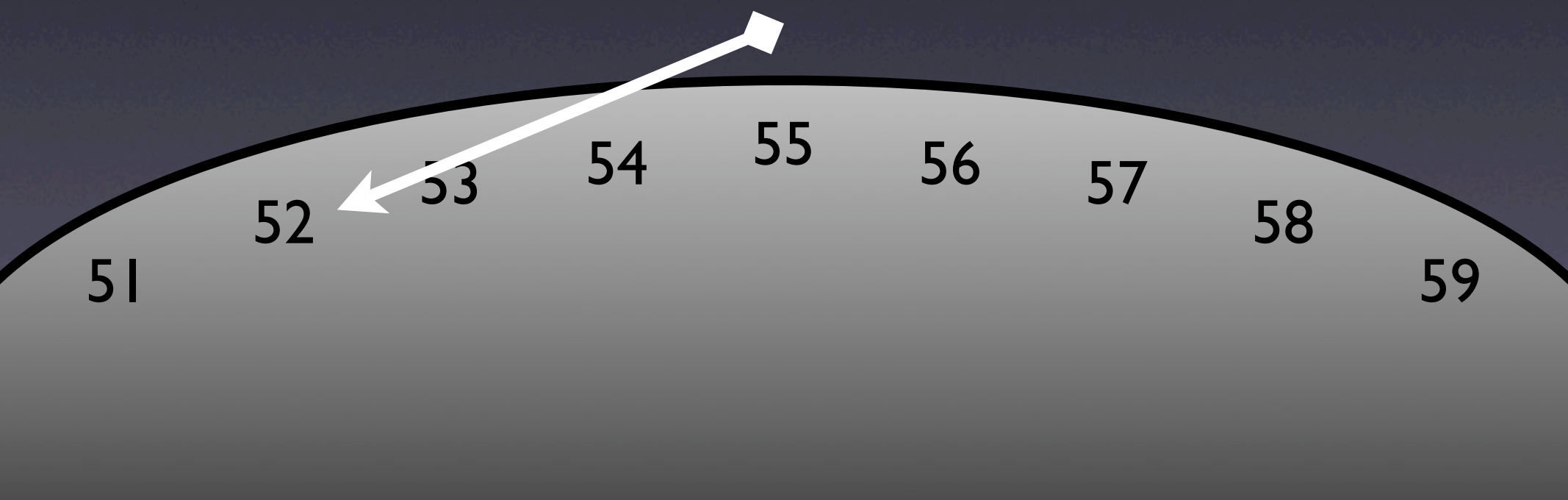
to send 0: Read @ 53
to send 1: Read @ 57



The Disk-arm Variant

LOW HIGH

{Read @ 52, Read @ 58}

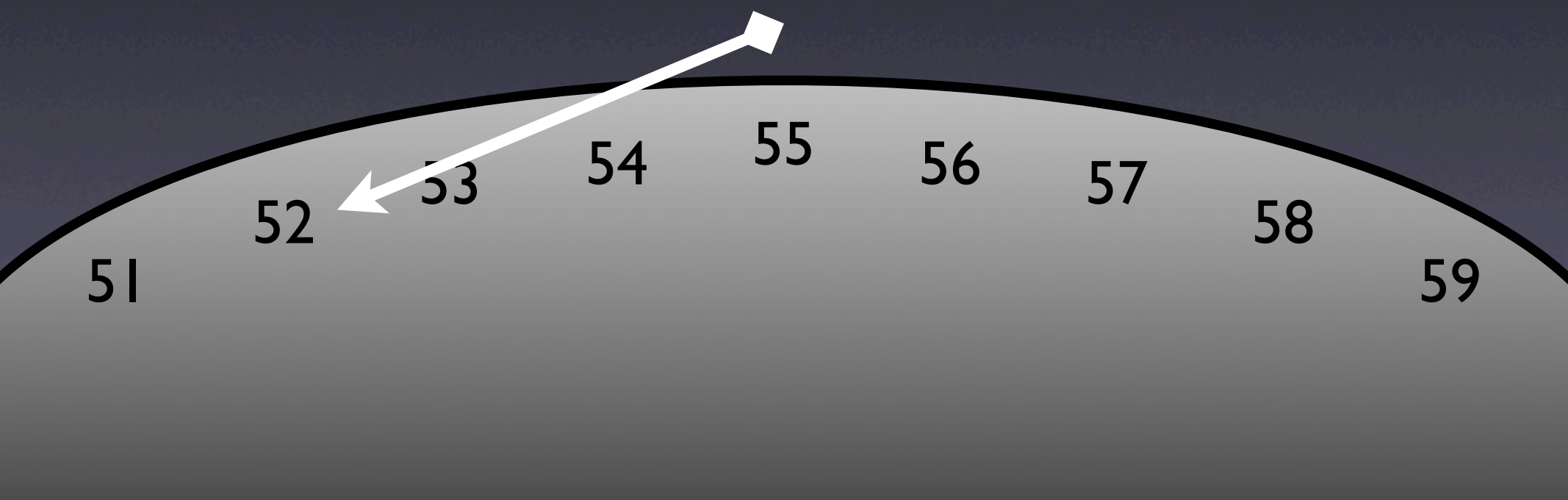


The Disk-arm Variant

LOW HIGH

Read @ 52 finished first

HIGH sent 0



Information Hiding

- Parties are allowed to communicate with the exception that content is:
  - censored
  - restricted to certain subjects
- e.g., hiding a message in the lowest order pixel bits of an image
  - referred to formally as steganography

Steganography

Cover Image

Image Containing Embedded Data

Similar techniques have been used to watermark copyrighted images

Subliminal Channels

- Introduced by the crypto community to circumvent US regulations (G J Simmons)
  - demonstrated channel using ElGamal and Schnorr signature schemes (EUROCRYPT '84)
- e.g., signal a bit by choosing one of two keys to sign a message

Where is this relevant?

- The use of covert channels is relevant in organizations that:
  - restrict the use of encryption in their systems
  - have privileged or private information
  - wish to restrict communication
  - monitor communications

Security Paradox?

- Otherwise strong security policies can be circumvented by covert methods
- Should we focus our attention towards making communication channels subliminal-free?
  - How would we do this?
  - We'll talk about this tomorrow

Network Covert Channels

- Information hiding
  - placed in network headers AND/OR
  - conveyed through action/reaction
- The goal is that the channel be undetectable or unobservable

Taxonomy

- Network covert channels can be
  - Storage-based
  - Timing-based
  - Frequency-based
  - Protocol-based
    - ...or any combination of the above

- Each of the above categories constitute a *dimension* of data
- Information hiding in packet payload is outside the realm of network covert channels
  - These cases fit into the broader field of steganography

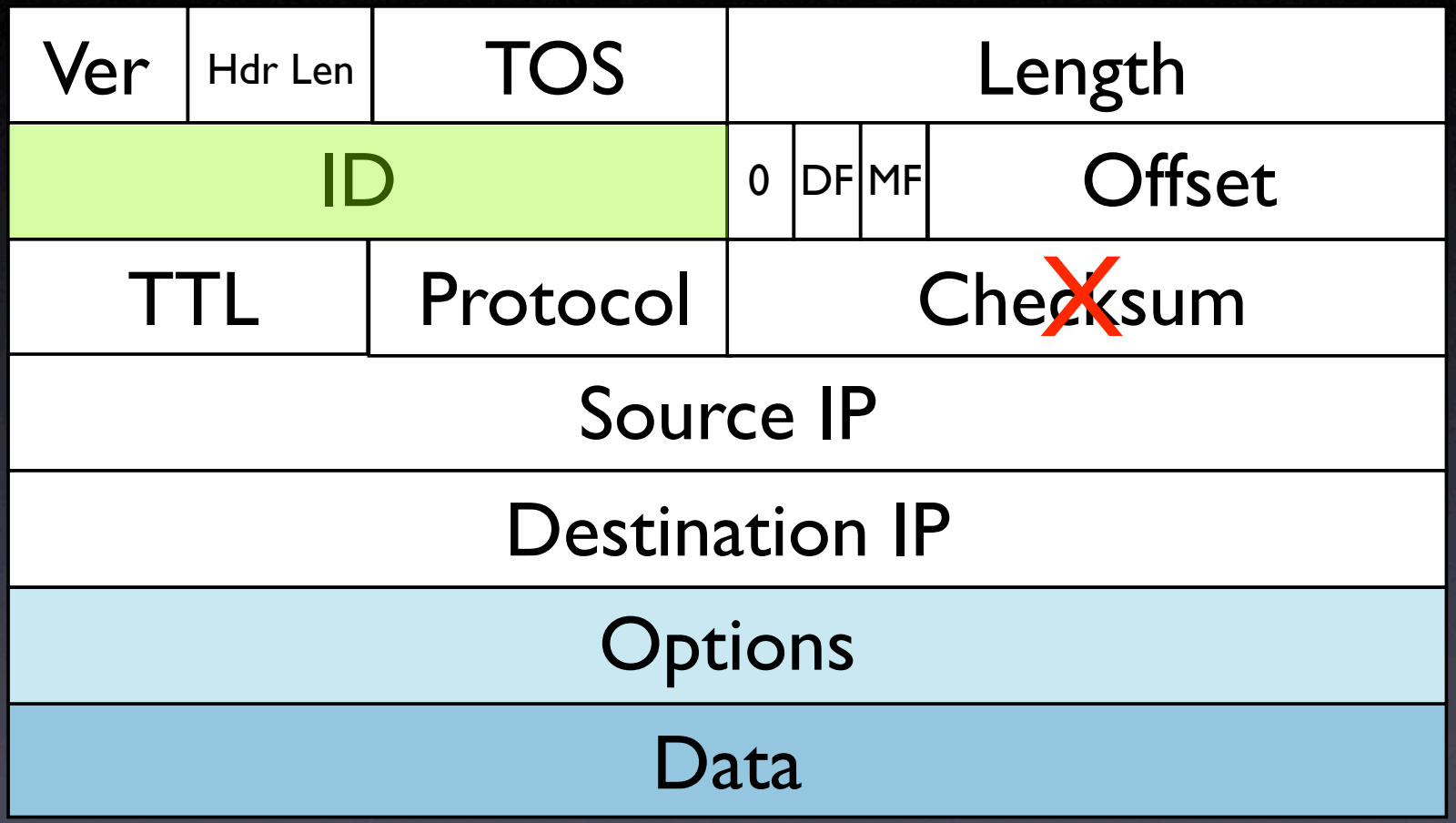
Aside: Steganography

- Steganography differs from covert channels in areas of
  - Interactivity
  - Persistence
  - Scope

Storage-based

- Information is leaked by hiding data in packet header fields
  - IP identification
  - Offset
  - Options
  - TCP Checksum
  - TCP Sequence Numbers

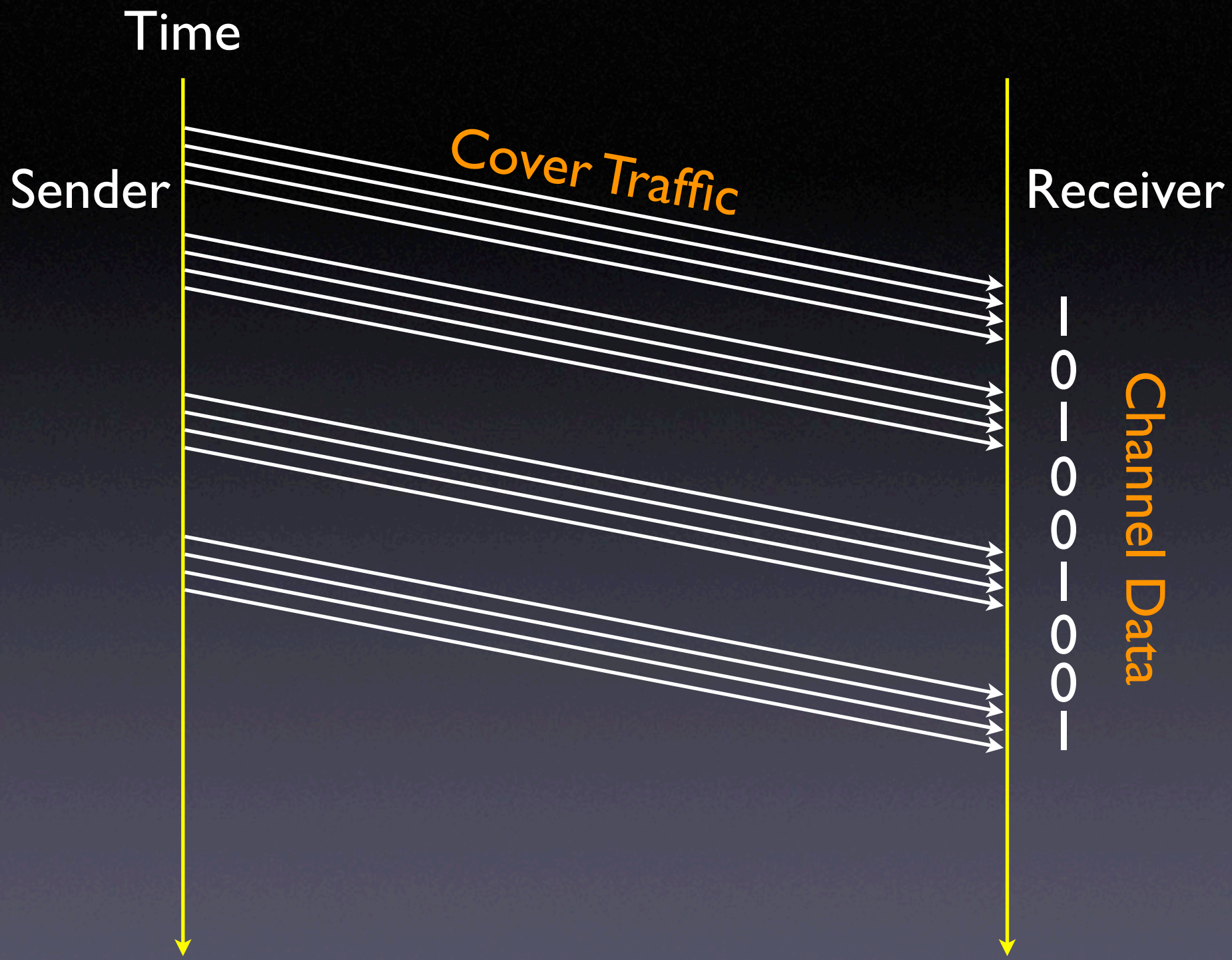
IP Header



* a limited number of options exist for these types of channels

Timing-based

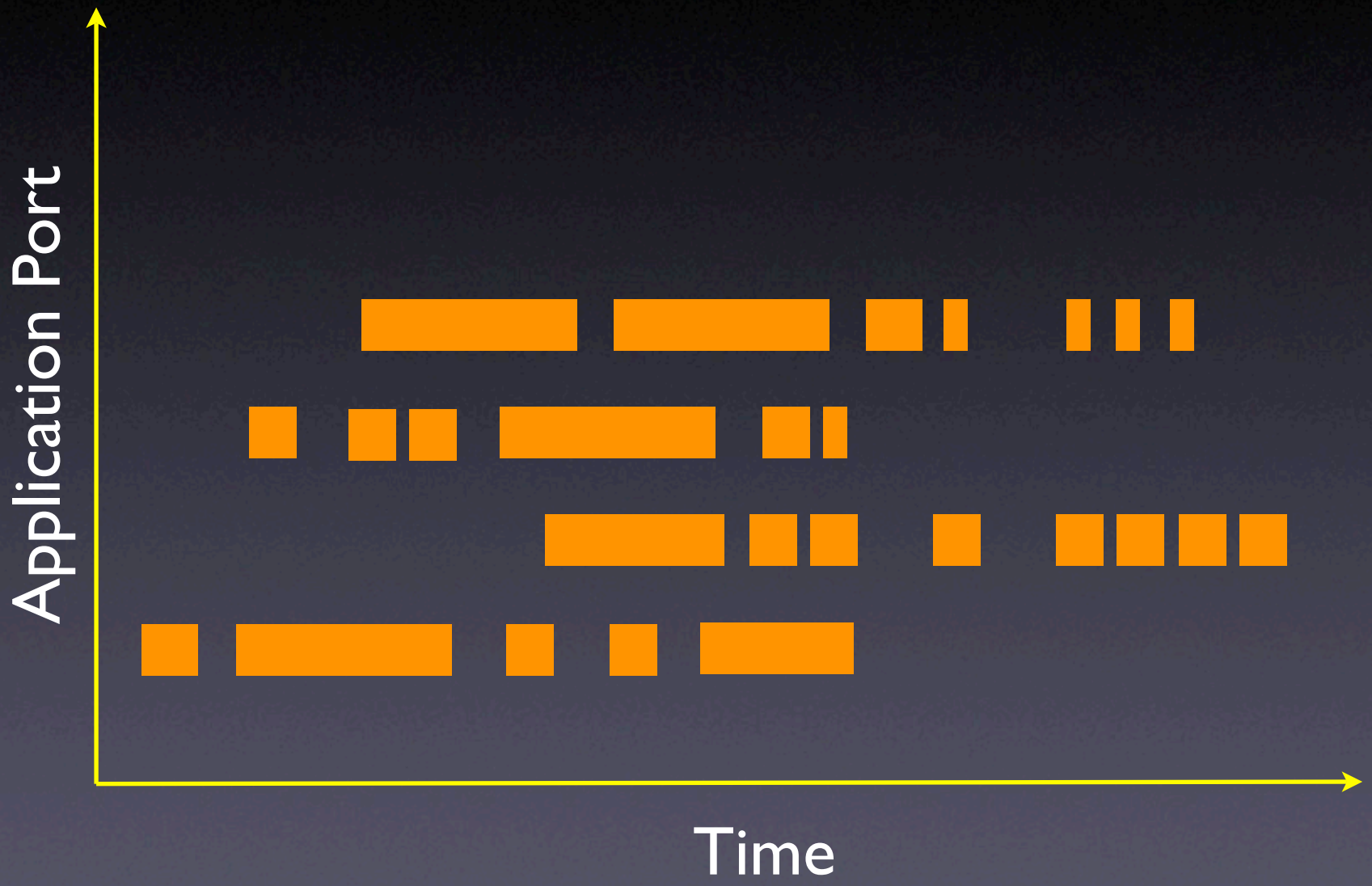
- Information is leaked by triggering or delaying events at specific time intervals
- The CBS channel is timing-based


Time
Sender
Cover Traffic
Receiver
1 0 1 0 0 1 0 0 1
Channel Data

Frequency-based

- Information is encoded over many channels of cover traffic using any of the other covert channel techniques
- The order or combination of cover channel access encodes information

Frequency-based



Protocol-based

- Exploits ambiguities or non-uniform features in common protocol specifications
- We'll see an example of this tomorrow

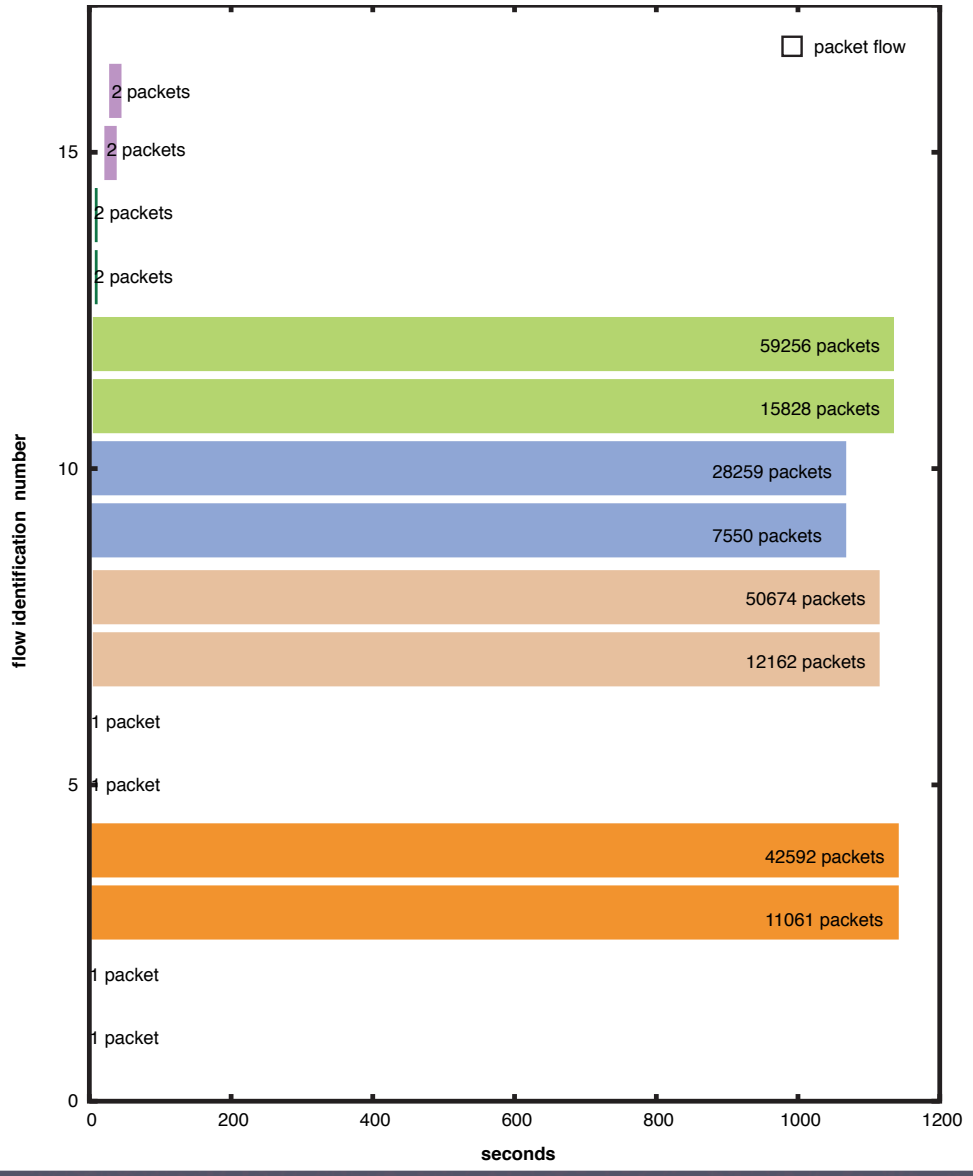
Threats to covert communication

- Discovery
  - Channel completely compromised
- Detection
  - Existence of particular channel is known
- Prevention Mechanisms
  - Proxy processing and delay added

Traditional Detection Mechanisms

- All detection is done through statistical methods
- Storage-based
  - Data analysis
- Time-based
  - Time analysis
- Frequency-based
  - Flow analysis

Flow Analysis



Packet Flow	# of Packets	% Packets
(4, 3)	42592	27%
(12, 11)	59256	33%
(8, 7)	50674	25%
(10, 9)	28259	15%

Pitfall: Randomness

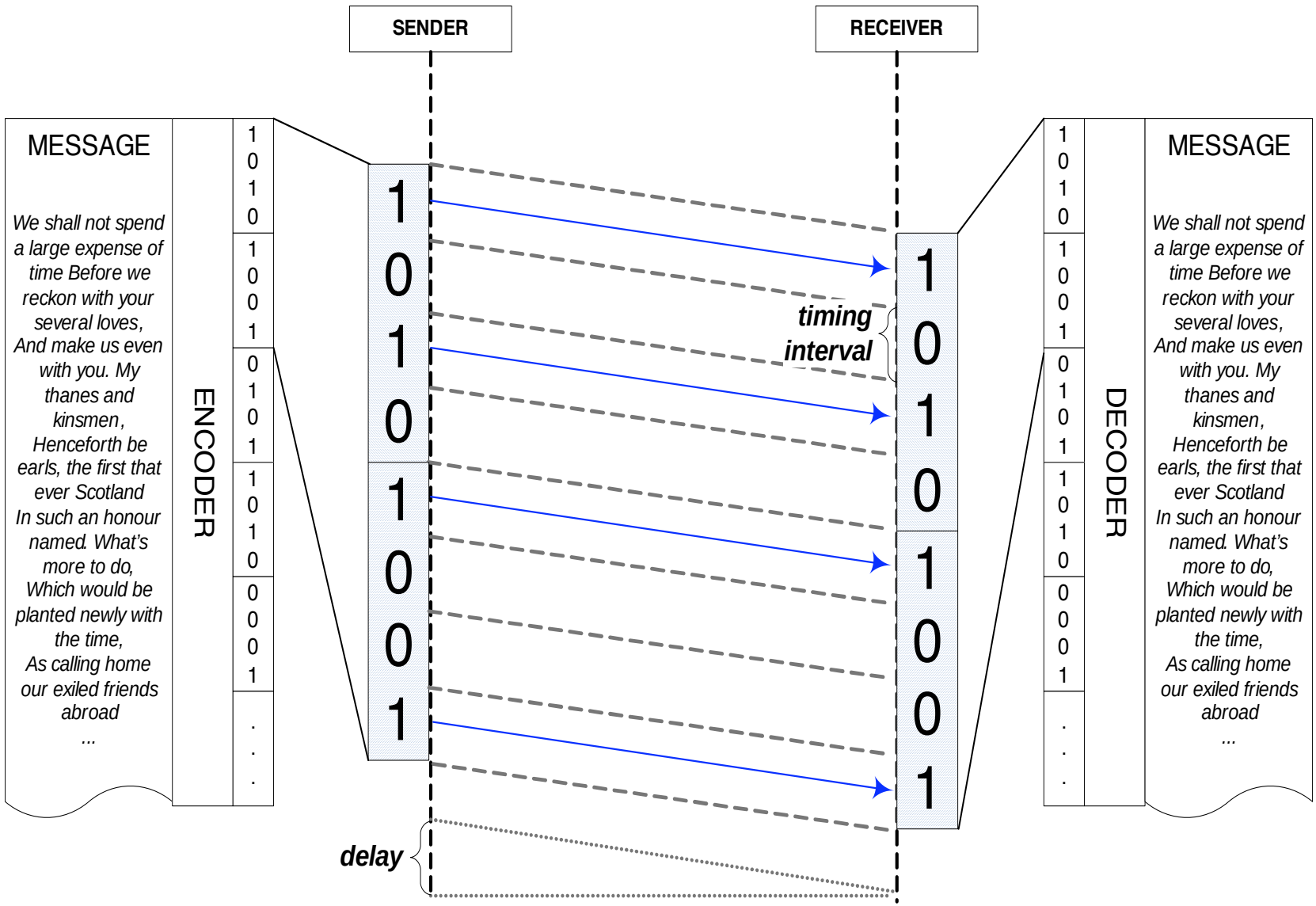
- In *Covert Messaging Through TCP Timestamps* [PET 2002] the authors use the low order bits of the TCP timestamp field to hide data
  - The authors encrypt their data before transmission to make the bits appear random
  - Oops, low order bits of TCP timestamp aren't cryptographically random [Defcon 10]

Implementation Pitfalls

- Synchronization and a priori setup
- Error-correction
- Feed-back
  - e.g., flow-control
- Side-effects
  - e.g., Symmetry, feature loss

The CBS Channel

- Developed by Cabuk, Brodley, and Shields
- Appeared at CCS 2004
- Covert data sent by traditional timing channel


SENDER
RECEIVER
MESSAGE
We shall not spend a large expense of time Before we reckon with your several loves, And make us even with you. My thanes and kinsmen, Henceforth be earls, the first that ever Scotland In such an honour named. What's more to do, Which would be planted newly with the time, As calling home our exiled friends abroad ...
ENCODER
1010
1001
0101
1010
0001
...
1010
1001
0101
timing interval
1010
1001
...
delay
1010
1001
0101
1010
0001
...
DECODER
MESSAGE
We shall not spend a large expense of time Before we reckon with your several loves, And make us even with you. My thanes and kinsmen, Henceforth be earls, the first that ever Scotland In such an honour named. What's more to do, Which would be planted newly with the time, As calling home our exiled friends abroad ...

The CBS Channel (2)

- Requires a priori knowledge for start event and timing interval
- Unidirectional channel
  - "No" feedback mechanism

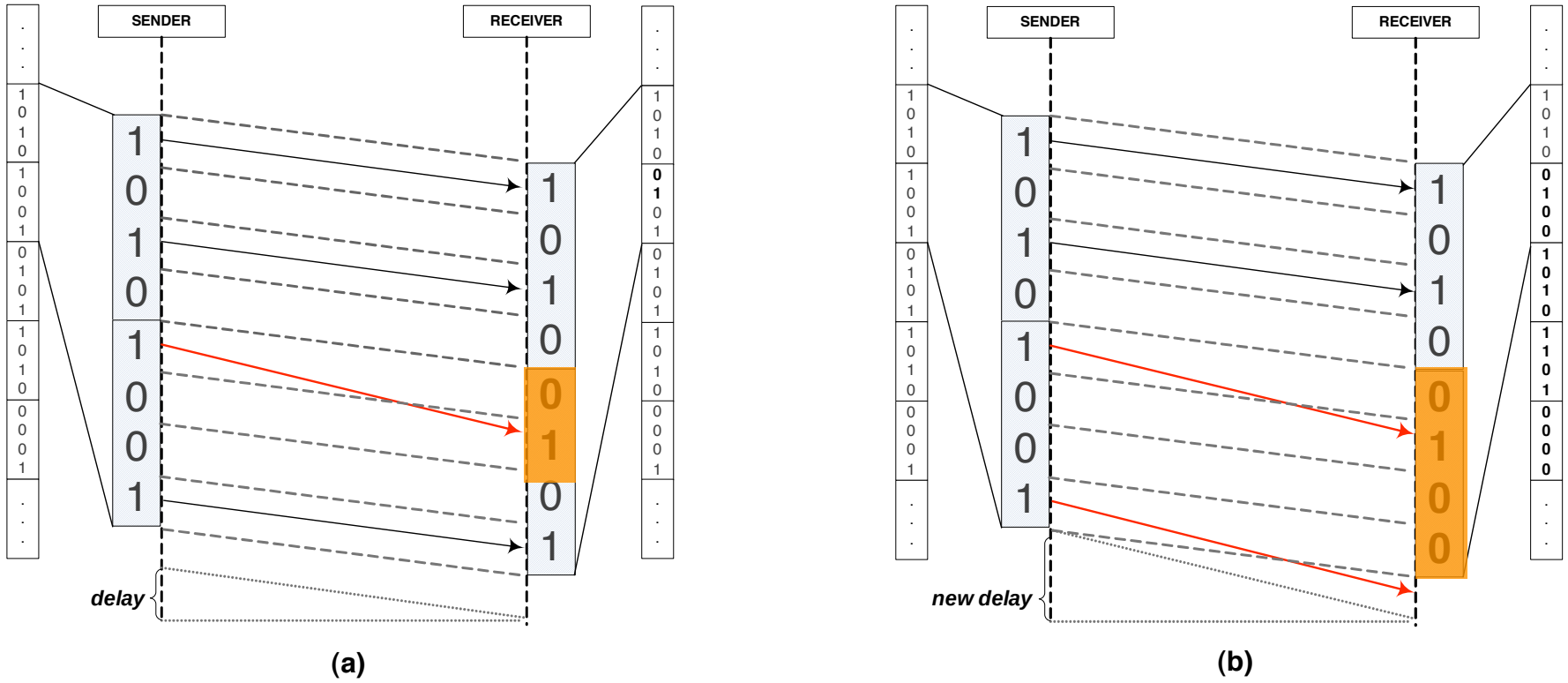
Performance Factors

- The following issues become important during implementation of the CBS channel
  - Network conditions
  - Sender/receiver processing capabilities
  - Algorithmic complexity

Determining the Time Interval

- The bandwidth of the covert channel is limited by:
  1. the processing speed of the hosts and network availability (upper bound)
  2. the length of the timing interval (lower bound)
- The goal is to have the smallest timing interval possible while retaining accuracy

Synchronization



2 bit flip

translation

Synchronization (2)

- Start of frame (SOF)
  - realigns clock synchronization between sender and receiver every few words
- Silent intervals
  - interval is decided a priori
  - allows sender some control over misaligned data

Interval Adjusting

- An additional mechanism to keep data intervals synchronized
  - Receiver computes delta as difference between *packet arrival* time and *expected arrival* time
  - Receiver adjusts timing interval by delta
    - Only works if delta is < 50% timing interval

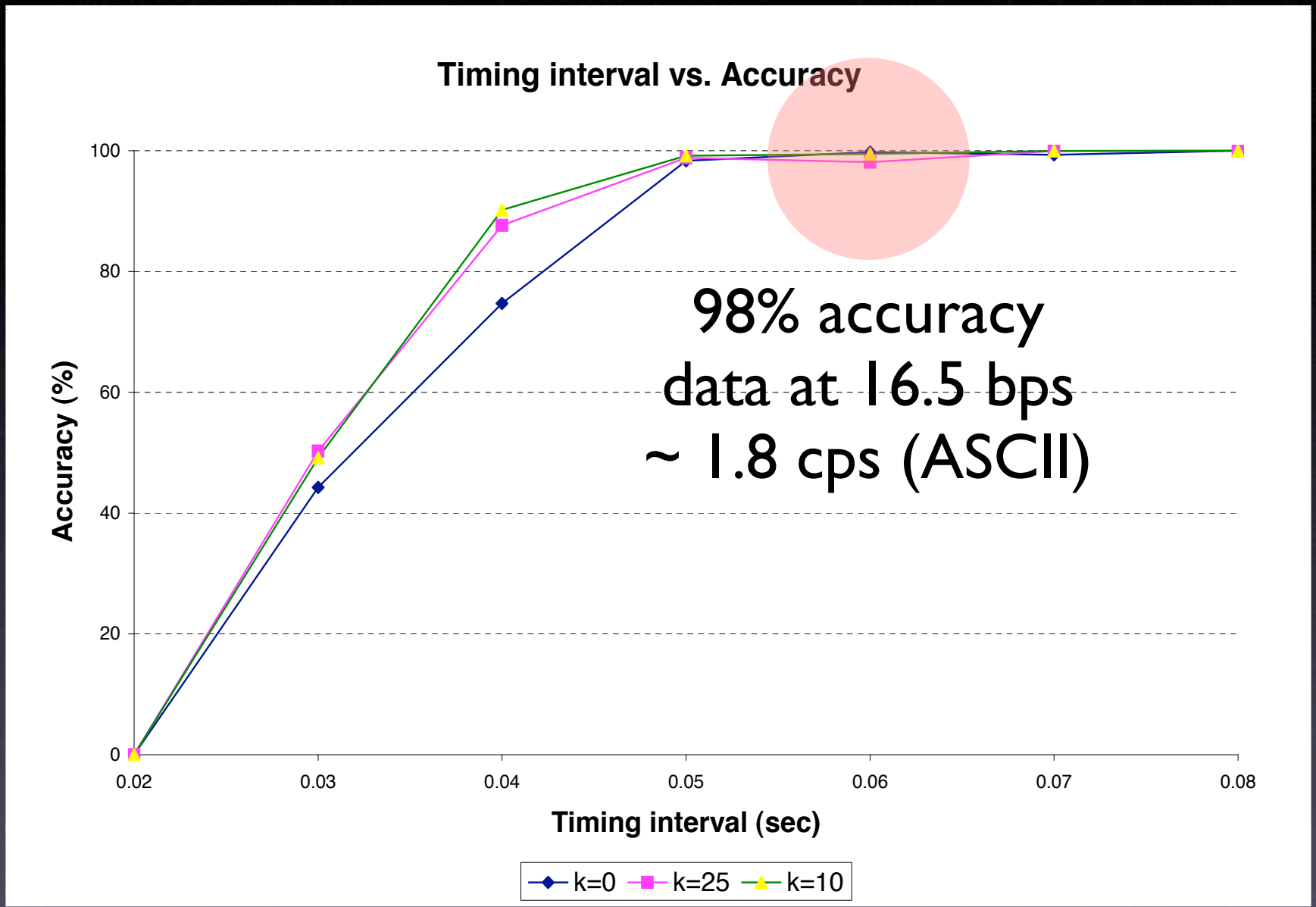
Evaluation of CBS Channel

- Accuracy of channel is based on *edit distance*
  - minimum distance between two strings needed to transform one into the other
    - e.g., "hassle" and "castle" have an edit distance of 2 characters

Purdue

RTT ≈ 31.5 msec
~12 hops

Georgetown


Timing interval vs. Accuracy
Accuracy (%)
100
80
60
40
20
0
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Timing interval (sec)
k=0
k=25
k=10
98% accuracy
data at 16.5 bps
~ 1.8 cps (ASCII)

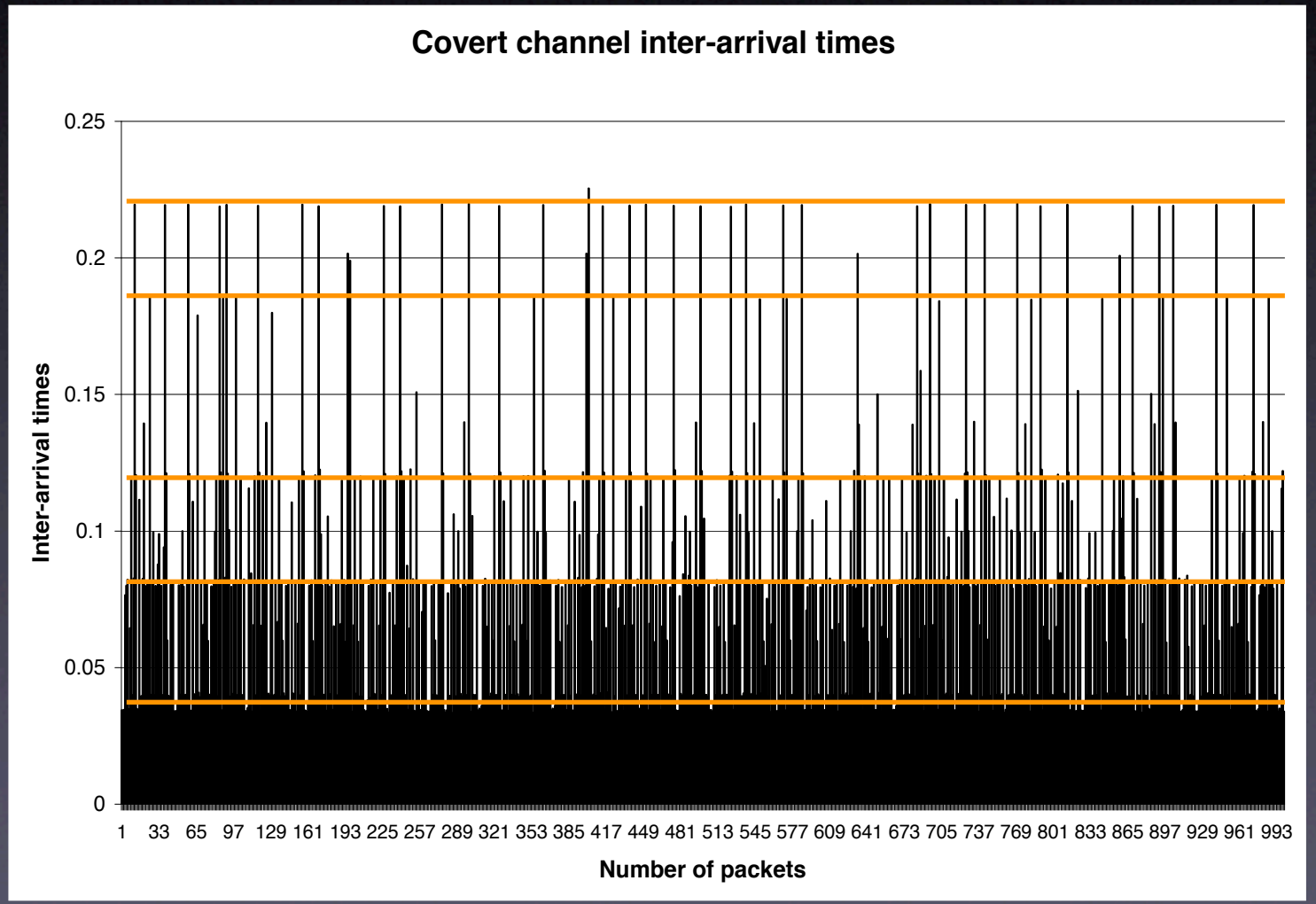
Effect of Network Conditions

- Channel run over congested network with high RTT variance
  - congestion lowers the accuracy
- Lesson:
  - *"... interval must be increased to retain accuracy during periods of high congestion"*

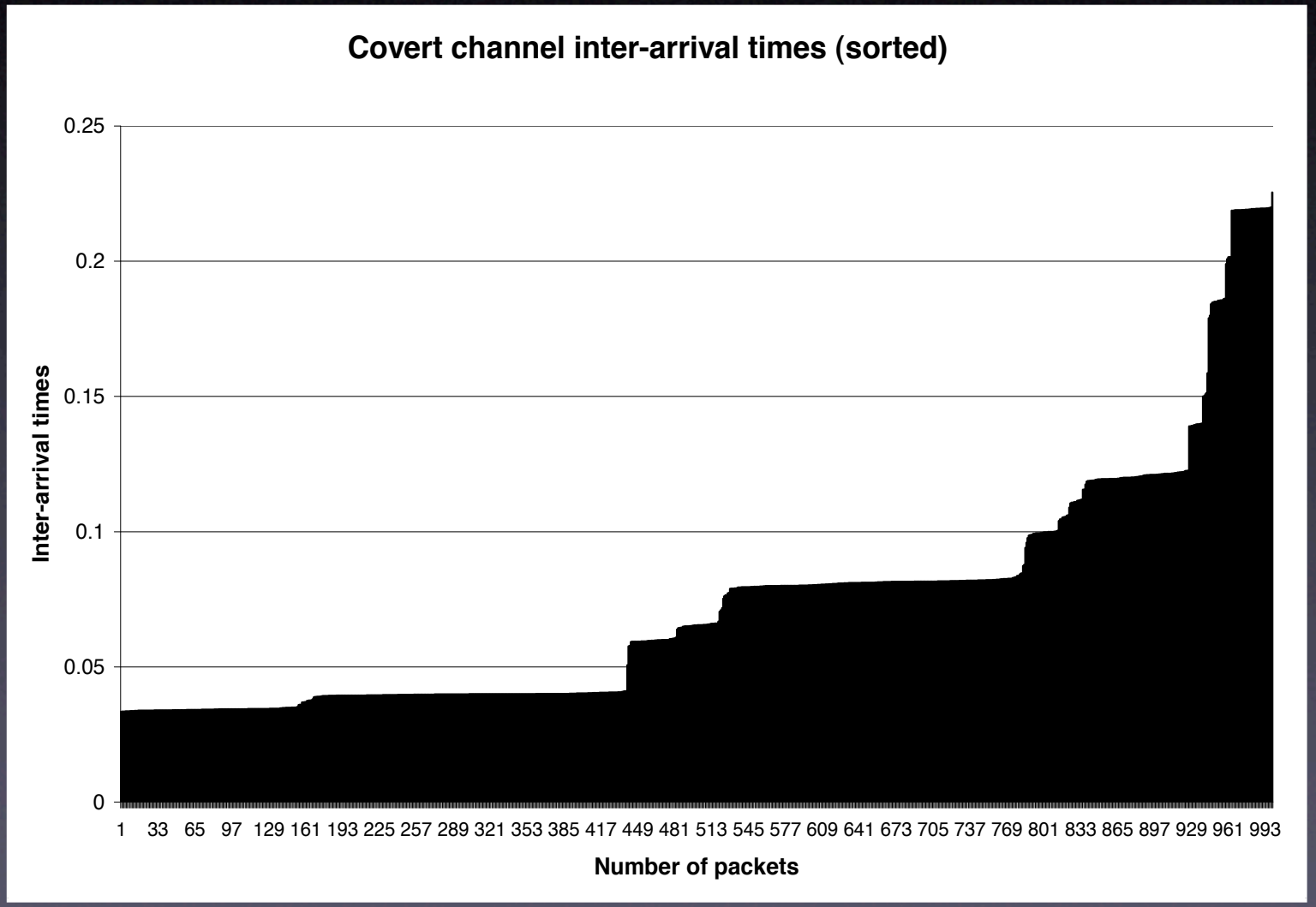
Detecting IP Covert Timing Channels

- Can we detect covert channels in IP traffic?

Detecting IP Covert Timing Channels



Detecting IP Covert Timing Channels (2)



Methods for Detecting Regularity in Inter-arrival (IA) Times

- A sample size of 2000 packets is used in these experiments
  - Measure 1: examine patterns in variance
  - Measure 2: ϵ-Similarity between adjacent inter-arrival times

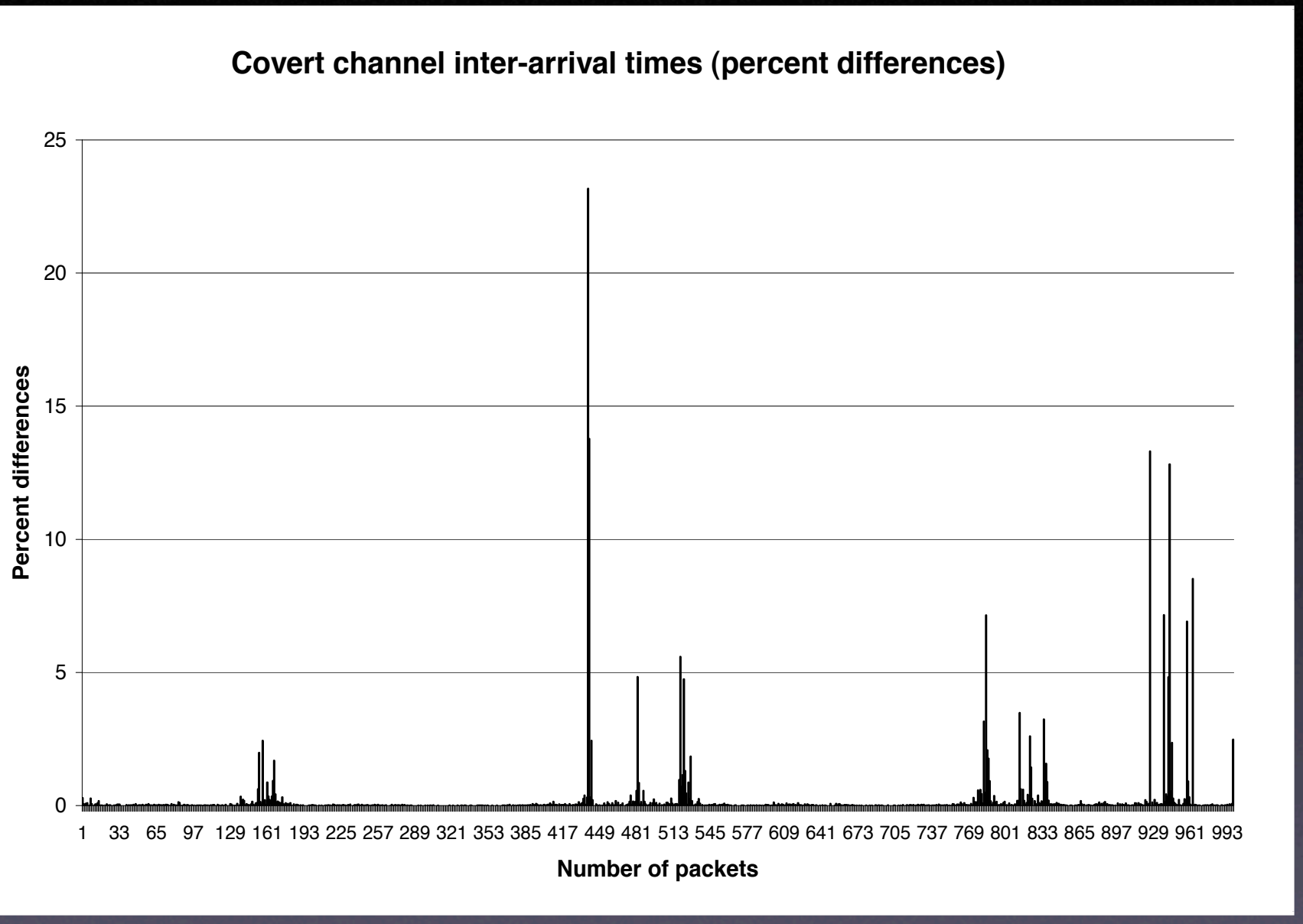
Patterns in Variance

1. Divide traffic into adjacent windows of size *w* packets
2. $\forall$ window *i* compute stdev σ_i of IA times
3. Calculate pairwise differences between σ_i and σ_j for all $i < j$
4. Calculate the stdev of the pairwise differences from (3), this is the metric of regularity

$$regularity = STDEV(\frac{|\sigma_i - \sigma_j|}{\sigma_i}, i < j, \forall i, j)$$

ϵ-Similarity between adjacent IA times

1. Using sorted IA times, compute the relative difference between consecutive points
   - $|P_i - P_{i+1}|/P_i$ for each point P_i and P_{i+1}
2. ϵ-Similarity is then computed as the percentage of relative differences less than ϵ
   - The pairwise difference is large only for jumps in the step function


Covert channel inter-arrival times (percent differences)
Percent differences
25
20
15
10
5
0
1 33 65 97 129 161 193 225 257 289 321 353 385 417 449 481 513 545 577 609 641 673 705 737 769 801 833 865 897 929 961 993
Number of packets

Empirical Evaluation of the Detection Metrics

- Three covert channels are used to test the efficacy of the aforementioned detection metrics

1. A simple timing channel
2. A timing channel with a varied time interval
3. A timing channel with manufactured noise

- What is the false negative rate for these methods in:

  1. covert channels?

  2. non-covert channels?

- Can these metrics be used to automate detection?

Data Sets

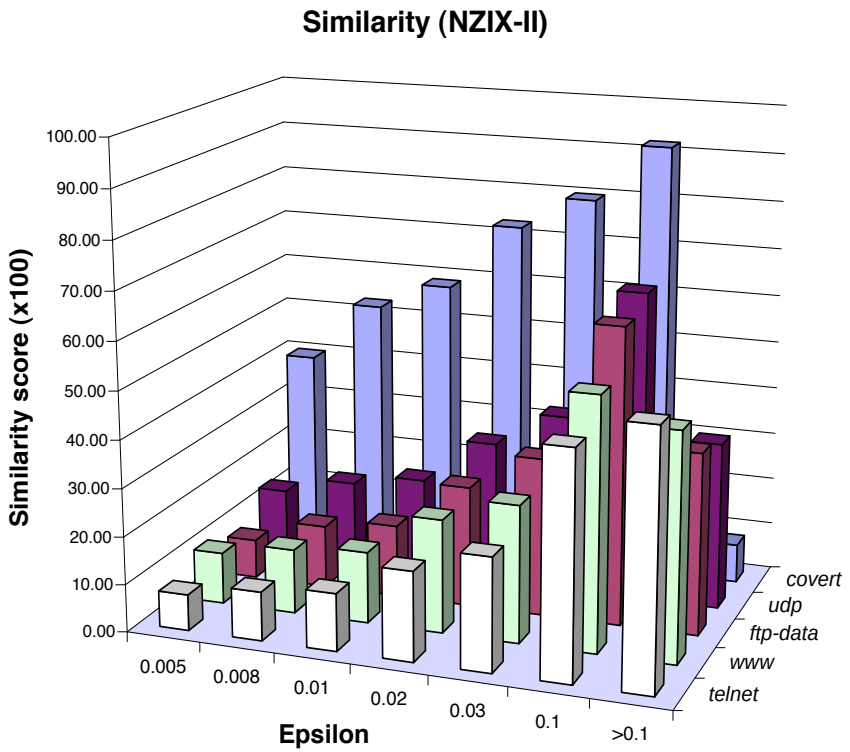
- '99 DARPA data set
  - Telnet and HTTP traffic
- NZIX-II data set
  - Telnet, HTTP, FTP, UDP traffic
- Used only flows that were ≥ 2000 packets
- Covert traffic data has a different scale of jitter than the trace data sets

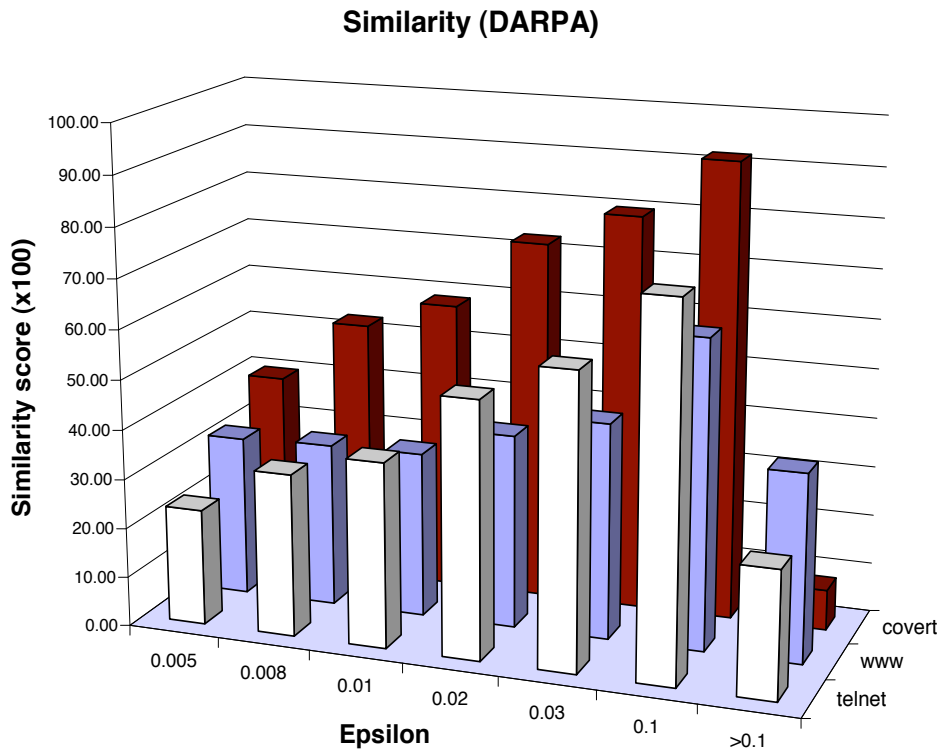
Variance Patterns for Simple Channel

Dataset	Application	w=250	w=100
NZIX-II	WWW	22.14	34.32
NZIX-II	FTPd	7.77	16.46
NZIX-II	Telnet	12.08	18.15
NZIX-II	UDP	16.57	27.18
DARPA	WWW	21.59	62.32
DARPA	Telnet	17.70	52.21
	Covert-I	2.18	4.63

Table 1: Regularity of NZIX-II, DARPA, and covert traffic with windows of size 250 and 100.

ϵ-Similarity for Simple Channel





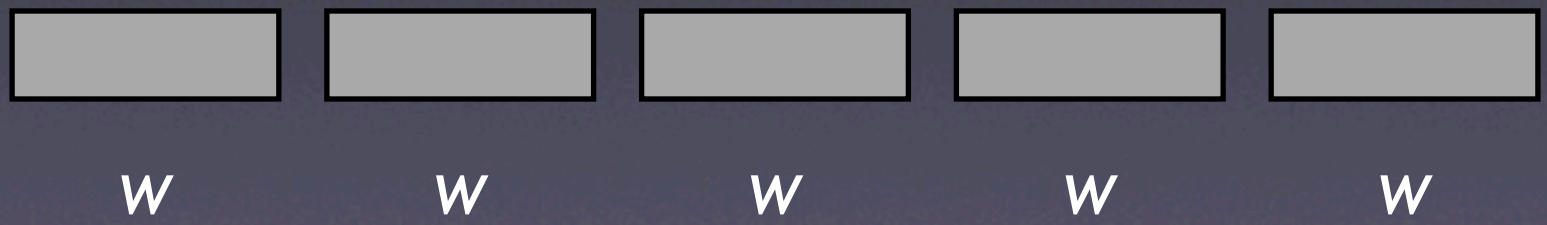
- 40% of the covert traffic has difference of less than $\epsilon = 0.005$
  - less than 15% of the non-covert traffic matches this difference
- The DARPA data appears far more regular than the NZIX-II data
  - the DARPA dataset was synthetically generated

Varying the Time Interval (Channel II)

- Motivation: obfuscate regularity
- Three interval values are chosen, cycled every *t* packets

Varying the Time Interval (Channel II)

- If *t* is larger than *w*, we can no longer compare the variance of windows (measure 1) to discover the channel
  - Even if *t* is much smaller than *w*, the variance between each window would be similar



ϵ-Similarity for Channel II

Method	t	ϵ-Similarity Score 0.005	0.008	0.01	0.02	0.03	0.1	>0.1
Sequential	250	34.17	45.17	51.23	67.38	75.29	90.75	9.25
	100	34.12	45.77	52.78	67.53	75.54	90.50	9.50
	50	34.22	46.87	53.68	67.68	75.09	89.89	10.11
	10	34.87	46.37	51.83	67.58	76.19	90.65	9.35
Random	250	36.51	48.02	53.47	68.30	76.20	90.49	9.51
	10	35.21	46.88	52.55	68.29	75.67	90.28	9.72
Original		39.92	52.83	58.58	72.79	79.74	91.85	8.15

Table 2: ϵ-Similarity scores for Covert Channel II. For each window of t packets, the interval is selected to be from the set (0.04, 0.06, 0.08). Results are shown for both selection methods (Sequential and Random) and for the original covert channel that employs a single interval (0.04).

ϵ-Similarity metric is robust for detecting channel II

Introducing Irregularity (Channel III)

- Insert portions of IA times from a non-covert traffic stream of the same cover type
  - e.g., if channel is run over port 80, use WWW traffic to inject noise at random blocks
    - Measure 1 fails because windows are no longer comparable
    - Measure 2 succeeds in distinguishing the channel

ϵ-Similarity for Channel III

Noise Level	Type of Noise	ϵ-Similarity Score						
		0.005	**0.008**	**0.01**	**0.02**	**0.03**	**0.1**	**>0.1**
0%		39.92	52.83	58.58	72.79	79.74	91.85	8.15
10%	WWW	36.54	47.50	52.67	66.46	73.39	87.46	12.54
10%	FTPd	35.03	46.05	51.30	64.89	71.45	84.94	15.06
10%	Telnet	34.89	45.83	51.14	64.29	70.70	83.17	16.83
25%	WWW	31.88	40.93	44.45	58.96	65.76	83.01	16.99
25%	FTPd	30.69	39.93	44.43	56.88	63.14	78.80	21.20
25%	Telnet	29.06	38.34	42.61	54.12	60.04	73.27	26.73
50%	WWW	31.70	37.31	40.33	53.15	59.52	79.32	20.68
50%	FTPd	26.12	32.21	35.60	46.35	52.39	70.53	29.47
50%	Telnet	24.21	30.31	33.31	42.47	47.72	61.40	38.60
Non-covert Traffic								
WWW		10.81	13.49	14.96	23.76	28.70	52.69	47.31
Telnet		7.54	10.25	12.04	18.69	23.65	46.99	53.01
FTPd		8.20	13.19	15.19	25.36	33.20	62.05	37.95

Table 3: ϵ-Similarity scores with different classes and levels of noise.

Automatic Detection of IP Covert Timing Channels

- Choose a threshold for each value of ϵ
  - values below ϵ are generated by covert traffic
  - threshold value is initialized by some number of training flows

False Positives in Automatic Detection

WWW	**Threshold**	FP	Cov-I	Cov-II	Cov-III(10%)	Cov-III(25%)	Cov-III(50%)
	$\mu + 2\sigma$	10.0	0.0	0.0	86.6	100.0	100.0
	$\mu + 1.5\sigma$	10.0	0.0	0.0	0.0	53.0	86.6
	$\mu + 1\sigma$	10.0	0.0	0.0	0.0	0.0	86.6
	$> Max$	10.0	0.0	0.0	0.0	20.0	86.6
FTPD	**Threshold**	FP	Cov-I	Cov-II	Cov-III(10%)	Cov-III(25%)	Cov-III(50%)
	$\mu + 2\sigma$	10.0	0.0	66.7	86.6	100.0	100.0
	$\mu + 1.5\sigma$	10.0	0.0	0.0	0.0	80.0	93.3
	$\mu + 1\sigma$	30.0	0.0	0.0	0.0	6.7	93.3
	$> Max$	10.0	0.0	0.0	0.0	33.3	86.6

Table 4: False positive (FP) and false negative (FN) rates for covert channel detection.

What about false negatives?

What have the authors told us?

- High bandwidth, undetectable covert channels are hard to make
  - as we reduce the bandwidth of our channel, we reduce the observability of the channel by statistical means
- We'll discuss our own conclusions tomorrow

Proposed Future Work

- Add error-correction
- Develop better synchronization techniques for increased channel bandwidth
- Investigate other detection methods for robust detection
  - What would you propose?

References

- IP Covert Timing Channels: An Initial Exploration - Cabuk, Brodley, Shields
- Covert Messages through TCP Timestamps - Giffen, Greenstadt, Litwack, Tibbetts
- An Analysis of Covert Timing Channels - Wray
- New Covert Channels in HTTP - Bauer

- Communication Using Phantoms: Channels in the Internet - Servetto, Vetterli
- Practical Data Hiding in TCP/IP - Ahsan, Kundur
- 20 Years of Covert Channel Modeling and Analysis - Millen