# (i) An approach to real-time wireless networking
# (ii) A clean slate approach to a secure wireless network: From axioms to protocols

P. R. Kumar
With I-Hong Hou and Vivek Borkar
With Jonathan Ponniah and Yih-Chun Hu

Dept. of Electrical and Computer Engineering
Texas A&M University

Email: prk@tamu.edu
Web: http://www.ece.tamu.edu/~prk/

ICTW
Maui
May 15, 2012

# From event-driven to time-driven computation

- Computers originally developed for computation
  - ENIAC (1946)
- Real-time computation (1973)
  - Digital control (circa 1960): Computation in feedback loop
- Hybrid systems (1990s)
  - Interplay of differential equations and logical dynamics
- Cyberphysical Systems

- How to support delay guarantees over an *unreliable* medium like wireless?
  - Goal: Formulate a mathematical *framework* for delay-based QoS

# Importance of providing latency guarantees: Wireless Tomorrow

- Current Internet
  - No guarantees – “Best effort”
  - At best – Throughput
- Increasing traffic with delay constraints
  - VoIP
  - Interactive Video
  - Cyberphysical systems
- How to support delay guarantees over an unreliable medium like wireless?

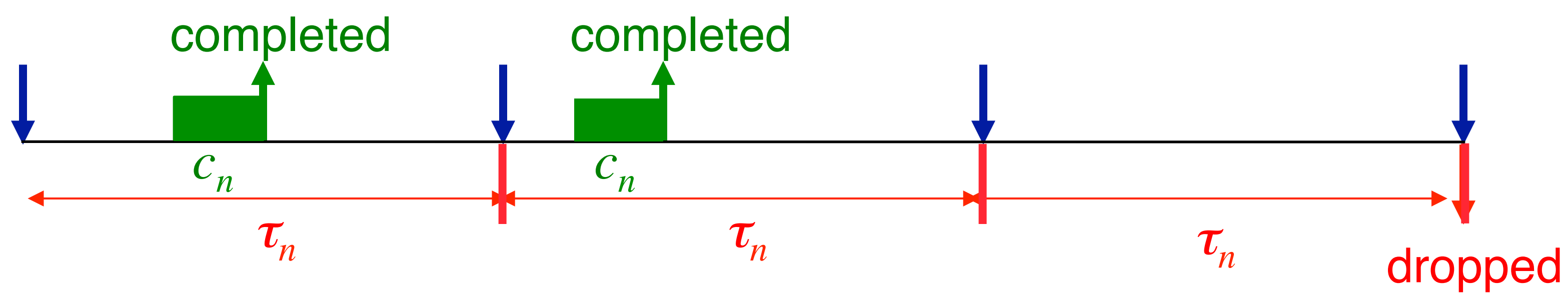
*In-Vehicle Networks*

- Wiring harness
  - Heavy
  - Complex
  - Costly

Replace wires by an access point

- Fewer mechanical failures
- Easier to upgrade

# Real-Time Scheduling: Liu-Layland (`73)

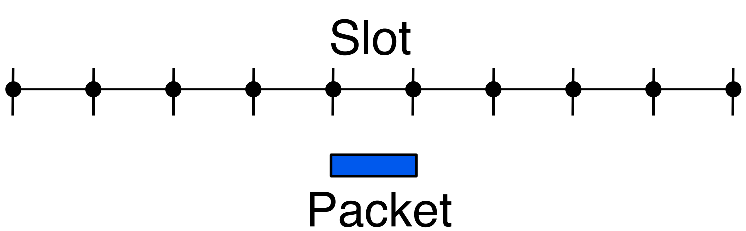



- $N$ tasks
  - Jobs of Task $n$ arrive with period $\tau_n$
  - Deadline is end of period
  - Worst case execution time $c_n$
- Rate monotone scheduling: Priority to smallest period task
- All deadlines met if $\sum_{n=1}^{N} \frac{c_n}{\tau_n} \leq N(2^{1/N} - 1)$ $\quad (\rightarrow \ln 2 = 0.69$ as $N \rightarrow \infty)$
- If any priority policy can meet all deadlines, then this policy can

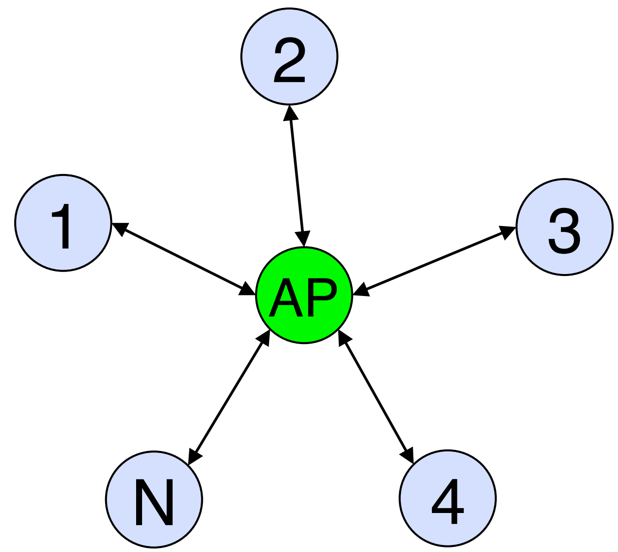
# Real-time communication

# Client-Server model

- A wireless system with an Access Point serving $N$ clients
- Time is slotted
- One slot = One packet



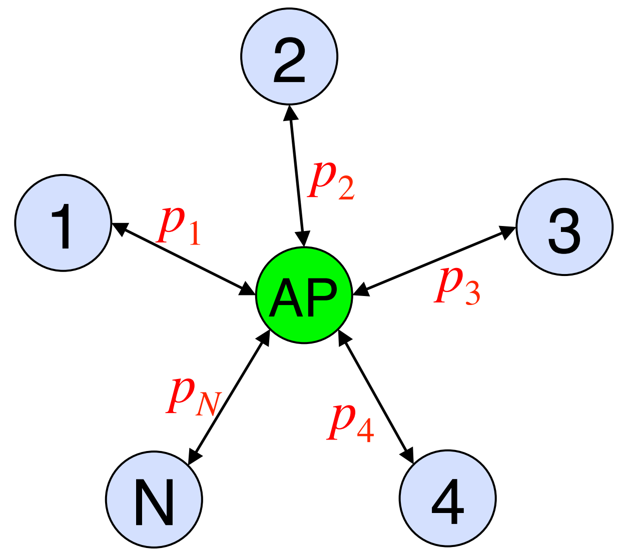

- AP indicates which client should transmit in each time slot



# Model of unreliable channels

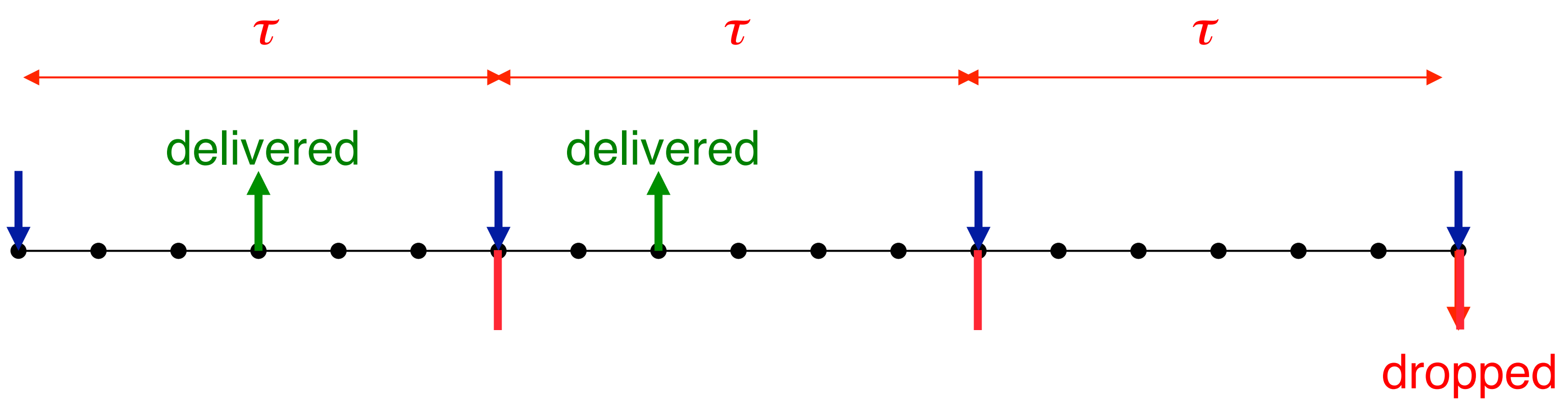
- Unreliable channels

- Packet transmission in each slot
  - Successful with probability $p_n$
  - Fails with probability $1\text{-}p_n$
  - So packet delivery time is a geometrically distributed random variable $\gamma_n$ with mean $1/p_n$




- Non-homogeneous link qualities
  - $p_1, p_2, \ldots, p_N$ can be different

# QoS model

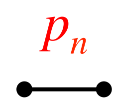



- Clients generate packets with fixed period $\tau$
- Packets expire and are dropped if not delivered in the period
- Delay of successfully delivered packet is therefore at most $\tau$
- Delivery ratio of Client $n$ should be at least $q_n$ packets/period

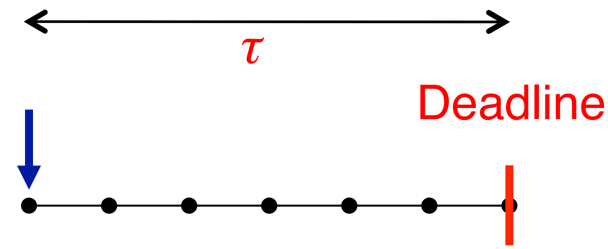
$$\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} 1(\text{Packet delivered to Client } n \text{ in } t\text{-th period}) \geq q_n \quad a.s.$$

# Multiple-time scale QoS requirements

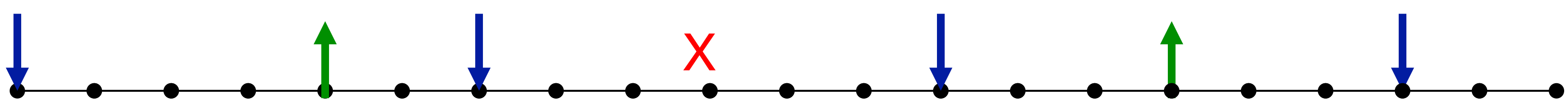
- Unreliable channels
  - Short time scale: Slots




- Arrivals and Deadlines
  - Medium time scale:
  - Period $\tau$ arrivals
  - Relative Deadline $\tau$




- Delivery ratio requirements
  - Long time scale:




$$\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} 1(\text{Packet of client } n \text{ delivered in } t\text{-th period}) \geq q_n \; a.s.$$

# Feasibility of a set of clients

# Load due to Client $n$

- The proportion of time slots needed by Client $n$ is

$$w_n = \frac{E(\text{\# deliveries per period}) \cdot E(\text{\# slots per delivery})}{\text{\# of slots of per period}}$$

$$= \frac{q_n \cdot \frac{1}{p_n}}{\tau}$$

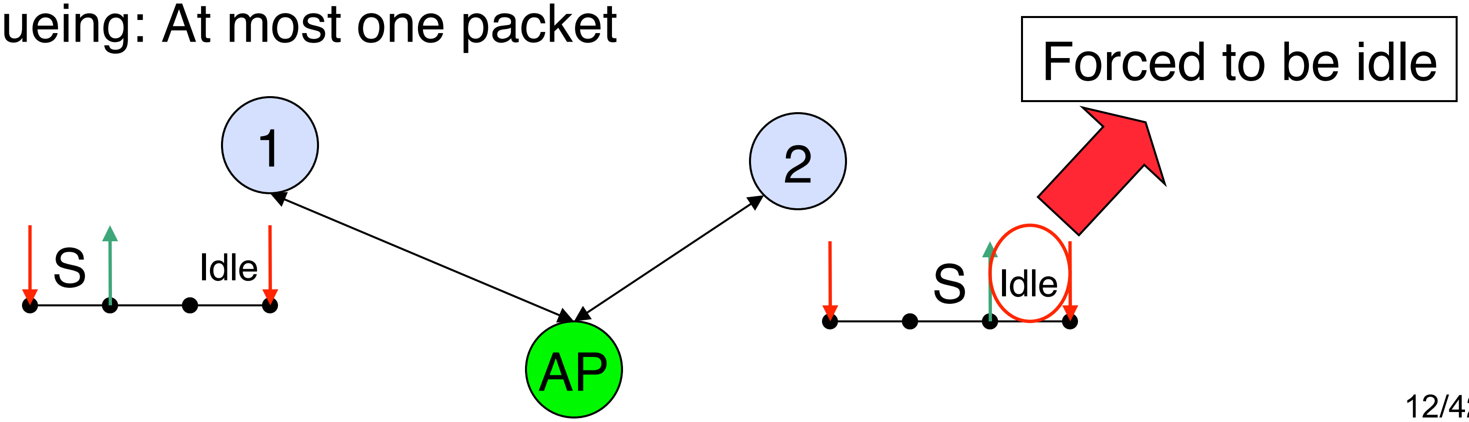
# Necessary condition for feasibility of QoS requirements

- Necessary condition from classical queueing theory

$$\sum_{n=1}^{N} w_n \leq 1$$

- Is it sufficient?
- No

- Reason: Unavoidable idle time
  - No queueing: At most one packet



# Stronger necessary condition

- Let $I(1,2,\ldots,N)$ := Unavoidable idle time after serving $\{1,2,\ldots,N\}$

$$I(1,2,\ldots,N) = \frac{1}{\tau} E\left[\left(\tau - \sum_{n=1}^{N} \gamma_n\right)^+\right] \quad \text{where } \gamma_n \sim \text{Geom}(p_n)$$

- Stronger necessary condition

$$\sum_{n=1}^{N} w_n + I(1,2,\ldots,N) \le 1$$

- Sufficient?

- Still not sufficient!

# Counterexample

- Two clients: Period $\tau = 3$
- Client 1
  - $p_1 = 0.5$
  - $q_1 = 0.876$
  - $w_1 + I_1 = 3.002/3 > 1$ ✗

$$w_1 = \frac{q_1}{p_1 \tau} = \frac{1.752}{3}$$

$$I_1 = \frac{\left(2p_1 + (1-p_1)p_1\right)}{3} = \frac{1.25}{3}$$

- Client 2
  - $p_2 = 0.5$
  - $q_2 = 0.45$
  - $w_2 + I_2 = 2.15/3 < 1$ ✓

$$w_2 = \frac{q_2}{p_2 \tau} = \frac{0.9}{3}$$

$$I_2 = \frac{1.25}{3}$$

- Clients {1,2}
  - $w_1 + w_2 + I_{\{1,2\}} = 2.902/3 < 1$ ✓

$$w_{\{1,2\}} = w_1 + w_2 = \frac{2.652}{3}$$

$$I_{\{1,2\}} = \frac{p_1 p_2}{3} = \frac{0.25}{3}$$

# Even stronger necessary condition

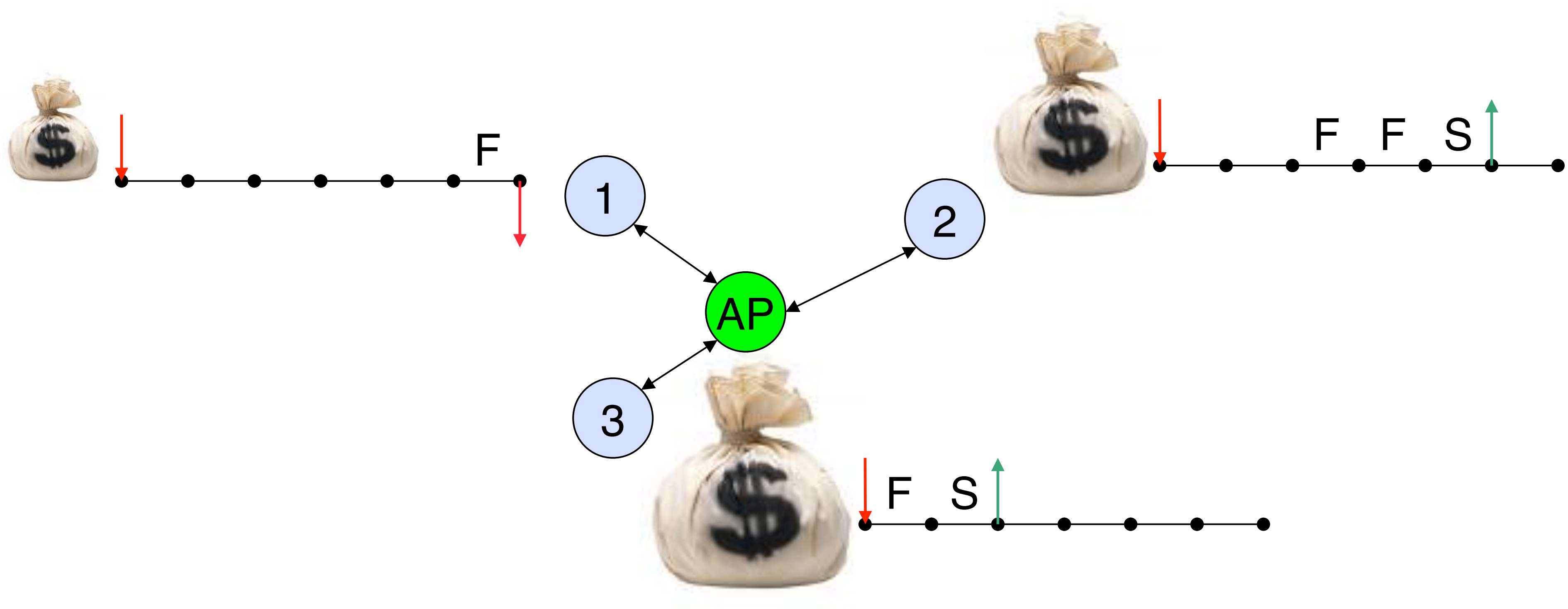
- Every *subset* of clients $S \subseteq \{1,2,\ldots,N\}$ should also be feasible
- Let $I(S) := \frac{1}{\tau} E\left[\left(\tau - \sum_{n \in S} \gamma_n\right)^+\right]$ = Idle time if only serving $S$
- Stronger necessary condition: $\sum_{n \in S} w_n + I(S) \leq 1, \quad \forall S \subseteq \{1,2,\ldots,N\}$

  $\underbrace{\nearrow \text{ with } S}$ $\underbrace{\searrow \text{ with } S}$

- Not enough to just evaluate for the whole set $\{1, 2, \ldots, N\}$
- **Theorem (Hou, Borkar & K ’09)**
  Condition is necessary and sufficient for a set of clients to be feasible

# Scheduling policy

# Debt-based scheduling policies




- Compute "debt" owed to each client at beginning of period
- A client with higher debt gets a higher priority on that period

# Two definitions of debt

- The **time debt** of Client $n$


$$= (w_n - \text{Actual proportion of transmission slots given to Client } n)$$

- The **weighted delivery debt** of Client $n$


$$= \frac{q_n - \text{Actual delivery ratio of Client } n}{p_n}$$

- **Theorem (Hou, Borkar & K ’09)**
  Both largest debt first policies fulfill every set of clients that can be fulfilled

# Computationally tractable policy for admission control

- Admission control consists of determining feasibility
- We need to check: $\sum_{n \in S} w_n + I_S \leq 1, \ \forall S \subseteq \{1,2,\ldots,N\}$
- Apparently $2^N$ tests, so computationally complex, but
- **Theorem (Hou, Borkar & K '09)**
  - Order the clients according to $q_n$ in decreasing order
  - Then we need only $N$ tests: Check $\{1, 2, \ldots, k\}$ for $1 \leq k \leq N$
  - $\{1,2,\ldots,N\}$ infeasible $\Leftrightarrow$ $\sum_{n=1}^{k} w_n + I(1,2,\ldots,k) > 1$ for some $k$
  - Polynomial time $O(N\tau \log \tau)$ algorithm for admission control

# Utility maximization for elastic traffic

# Utility maximization framework

- Client $n$ has a utility function $U_n(q_n)$
  - $U_n$ positive, str incr, str concave, $U_n(0)$ = right limit …
- Maximize the total utility

- SYSTEM

Max

$$\sum_n U_n(q_n)$$

s.t.

$$\sum_{n \in S} \frac{q_n}{\tau p_n} \leq 1 - I_S, \forall S$$

over

$$q_n \geq 0$$

Solving SYSTEM directly is difficult

Clients may have different utility functions $U_n$

$2^N$ feasibility constraints

# Two sub-problems

Considers own utility function

**Client $n$**

$$\underset{0 \le \rho_n \le \psi_n}{Max}\ U_n\left(\frac{\rho_n}{\psi_n}\right) - \rho_n$$

Price $\psi_n$ ←

→ Payment $\rho_n$

Considers feasibility

**Access Point**

$$\underset{\{q_n\}\ \text{feasible}}{Max} \sum_{n=1}^{N} \rho_n \log q_n$$

Achieved by Weighted Transmission Time Policy

$u_n(t)$ = Number of slots in $[0, t]$ given to Client $n$

Give priority to lowest $u_n(t)/\rho_n$

No one needs to know channel reliability

Is weighted max-min fair
And weighted proportionally fair

**(Hou & K ’09)**

# Concluding remarks

- A *framework* for delay-based QoS that encompasses
  - deadlines
  - channel unreliabilities
  - timely throughput
  - client utilities
  - fading channels
  - correlated arrivals
  - rate adaptation
  - minimum throughput requirements,
  - broadcasting (network coding), etc.
- Analytically tractable
- Implementable policies
- Approach to *real-time wireless networking?*

# A clean slate approach to a secure wireless network: From axioms to protocols

P. R. Kumar

With Jonathan Ponniah and Yih-Chun Hu

# Goals

- Principled approach to security
  - Holistic design of security
  - Security is not an afterthought
- Security first, performance second
  - Performance subsequently optimized while preserving security
- Reverse of the usual approach

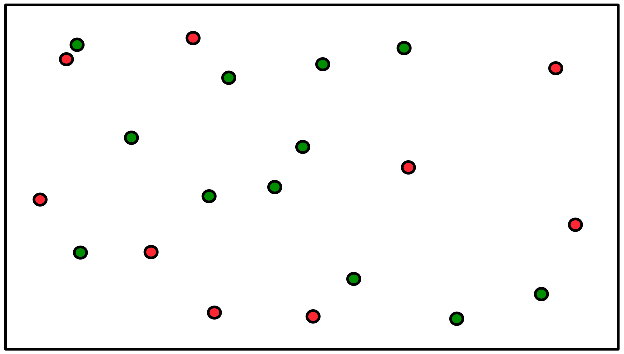
- Clean slate design of secure wireless networking
- Provably secure design
- Max-Min Optimal
- Complete suite of algorithms/protocols
- (Run applications based on temporal coordination)

A lot of explanation is clearly needed …

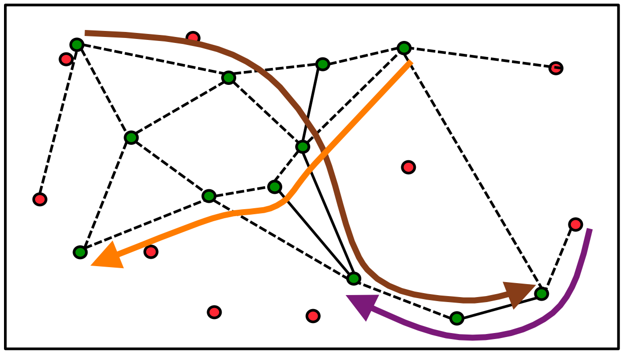
# Basic objective

- A complete suite of algorithms/protocols that takes you

- From startup
  - With just a set of nodes
  - Some good
  - Some bad
  - Good nodes don't know who the bad nodes are

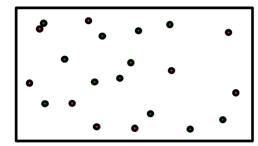


- To an optimized functional network carrying data reliably



# What can go wrong with a network formed in presence of bad nodes?

- Some nodes are bad. What can go wrong?



- Lots of things. A bad node could
  - Refrain from relaying a packet
  - Advertise a wrong hop count
  - Advertise a wrong logical topology
  - Jam
  - Cause packet collisions
  - Behave uncooperatively vis-à-vis medium access
  - Disrupt attempts at cooperative scheduling
  - Drop an “ACK”
  - Refuse to acknowledge a neighbor’s handshake
  - Behave inconsistently

“Byzantine” behavior

# One approach

- Identify "ATTACKS"
- Provide "DEFENSES"
- Result is
  - A sequence of patches
  - Arms race

- Issue
  - What other attacks are possible?

- Can we come up with provably secure architecture?
- Principled design: Holistic approach to security, not afterthought
- Complete suite of protocols from start-up to reliable operation

# Main results

- Protocols that lead from start-up to functional policed system

- Resulting network is Min-Max optimal with respect to utility

$$\underset{\text{All behaviors of bad nodes}}{Min} \ \underset{\text{Protocols}}{Max} \ U(x)$$

- In fact we will show

$$\underset{\text{Bad nodes can choose to either Jam or Cooperate}}{Min} \ \underset{\text{Protocols}}{Max} \ U(x)$$

- Hence, bad nodes are restricted to following actions
  – Either Jam or Cooperate in a consistent way

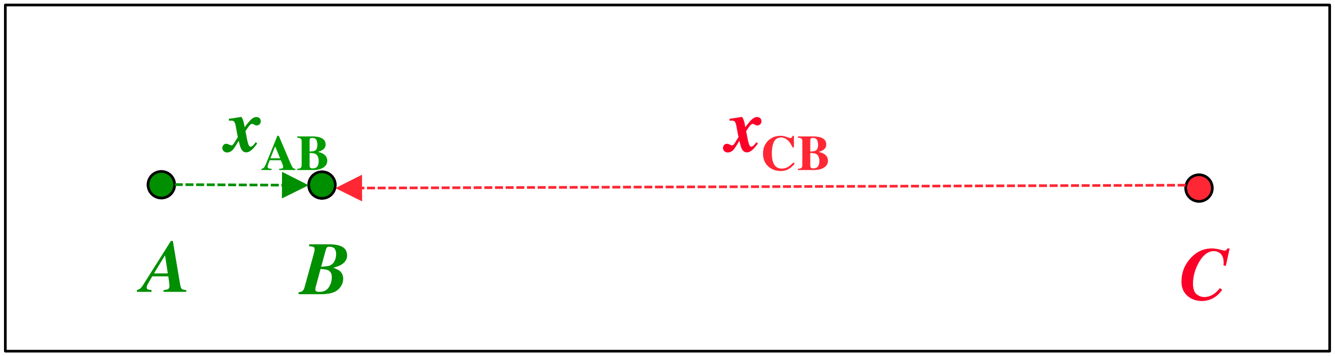
- (Also, timings applications can be consistently run on network)

# Implications of results

- Bad nodes can either choose to Jam or Cooperate
  - In a way that is consistent for each concurrent transmission set
- Nobody can prevent Jamming or Cooperating when it is done in a consistent way
- Other more malicious behaviors are ruled out
  - Not relaying a packet
  - Dropping an ACK
  - Presenting a wrong logical topological view
  - Disrupt medium access cooperation
  - Disrupt timing applications by inconsistent behavior
  - Not cooperating, disrupting, lying, spreading rumors, etc

# Why would a bad node ever cooperate?

- $U(x) = \mathrm{Min}(x_i)$




- If C jams, it can reduce $x_{AB}$

$$\lim_{|BC| \to \infty} x_{AB} = x_{AB}^{Max}$$

- If C pretends to be good, it can insist on equal share

$$\lim_{|BC| \to \infty} x_{AB} = 0$$

# Limitations and extensions under study

- Approach is not information-theory based
- It is packet based
- In particular, probabilistic unreliable channel is abstracted as a reliable channel of lesser rate
  - "Rate Adaptation"
- Issues of attacking this very abstraction are not addressed here today

# Fundamental ingredients of our approach

- Standard cryptographic primitives are assumed
  - All packets are encrypted
  - Bad nodes cannot create fake packets, cannot alter good packets without getting caught, etc

- And, importantly,

- Clocks and synchronization

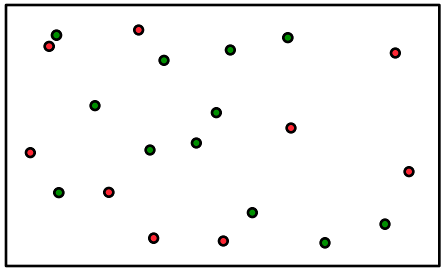
# Why clocks and synchronization?

- Without a notion of *time*, we cannot even talk of throughput
  - Without throughput we cannot talk of network Utility
- So *time* is an essential ingredient

- Without a notion of *common* time, nodes cannot cooperate temporally
  - They cannot share resources in a time-based way
  - Cooperative scheduling, etc., will be impossible

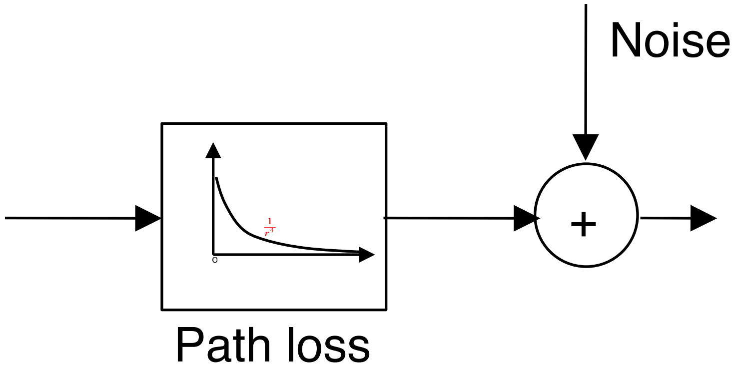
- So *synchronization* will be a fundamental ingredient
  - Facilitates temporal cooperation

# Technical Assumptions (approximately) (1)



- Bounded domain
- $n$ nodes, some bad
- Minimum distance between any pair of nodes
- Nodes are not mobile

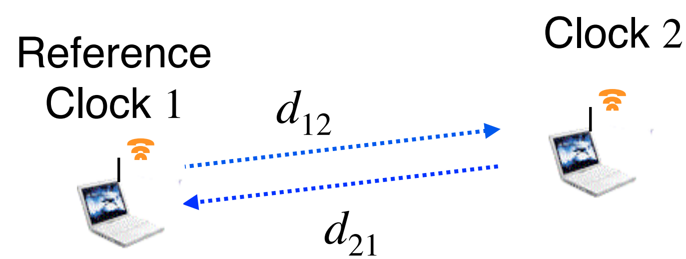
- Max power constraint at each node
- Noise at each node



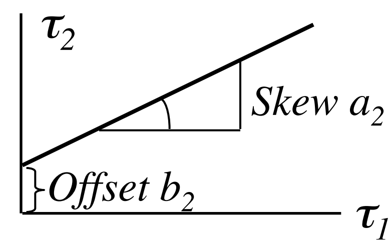

- Path loss is a function of distance
- SINR based rate

- Half-duplex nodes (can relax this)

# Technical Assumptions (2)



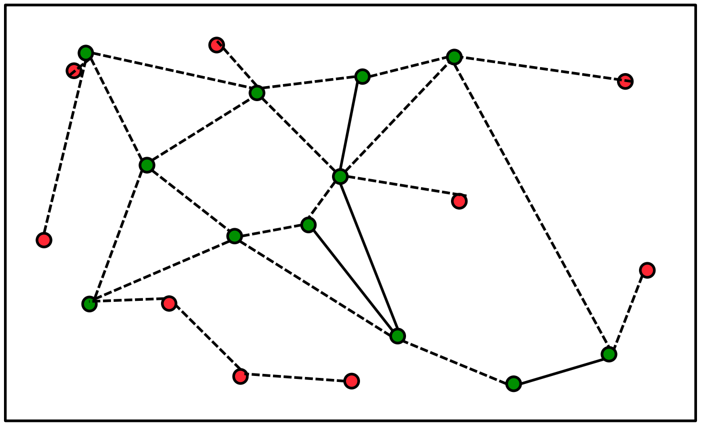

- Affine clock at each node
  - $0 < 1- \varepsilon \leq \mathrm{Skew} \leq 1 + \delta$ for all nodes




- Packets take a delay $d_{ij}$ from node $i$ to node $j$
- Each node has a private key
- Each node has a certificate which binds a public key to its identity and that is signed by a trusted authority

# Technical Assumptions (3)

- Assumption on connectedness
  - Suppose all nodes transmit at Max power
  - Then suppose there is an edge between each pair of nodes $(i, j)$ an for which $\text{SINR}_{ij}$ and $\text{SINR}_{ji}$ both exceed $\text{SINR}_{\text{threshold}}$
  - Assumptions
    - » Resulting graph is connected
    - » Subgraph of good nodes is also connected



# The Approach and Some Issues

- Nodes need to discover who their neighbors are.
  - Require a two-way handshake between the nodes.
  - How can we guarantee that any two nodes can communicate packets with each other when other nodes are liable to transmit at the same time and cause collisions?
  - Need an orthogonal medium access scheme.
  - Must operate with clocks that are not synchronized but also tick at different and unknown rates.
  - Nodes will need to synchronize their clocks with neighbors.
- Nodes will need to synchronize their clocks with neighbors.
  - Fundamental limitations to clock synchronization
  - Nodes can synchronize their skews but not their offsets which are indistinguishable from delays.

# The Approach and Some Issues (2)

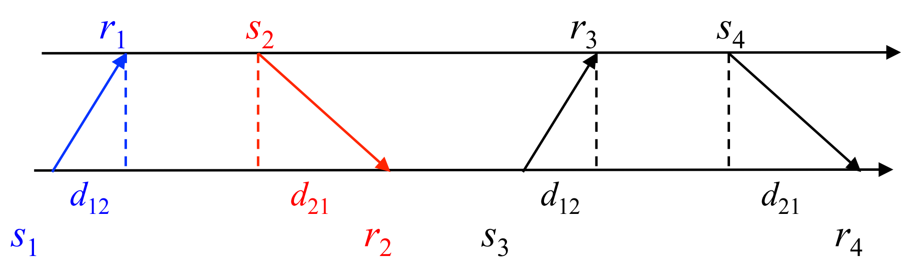
- Nodes need to form a network.
  - Require network wide consistency checks
  - Everything has to be done in the presence of malicious nodes
- Nodes draw up a schedule for transmissions and send data.
  - Some malicious nodes that conformed hitherto or remained hidden hitherto may not cooperate.
  - This requires a check to detect malicious behavior and another round of network wide computation with the un-cooperating nodes being taken into account.

# The Approach and Some Issues (3)

- All this also has to be done with a  finite bound on clocks and in the presence of skew errors
- Some challenges when we also aim for ε-optimality over network lifetime.

# Phases of protocol (1)

- Neighbor Discovery Protocol
  - Use orthogonal MAC codes
  - Within a bounded time all nodes discover their neighbors
- Clock Synchronization Phase
  - Pairs of neighboring nodes synchronize clocks

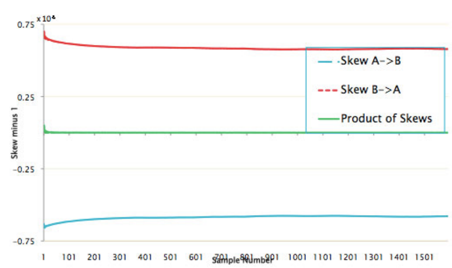


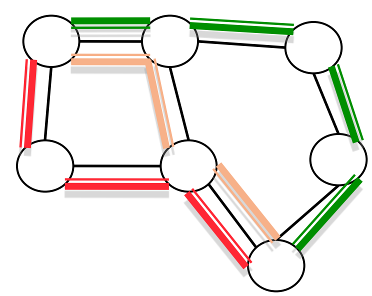

    - » Skews can be determined
    - » Offsets and one-way delays *cannot* be identified
    - » Round trip delays are determined
    - » They obtain capability to predict when packet reception times
    - » They also identify and certify each end-point and certify state of link

# Phases of protocol (2)

- Route discovery phase
  - Nodes flood the link-states throughput the network

- Consistency check phase
  - Nodes check that for all cycles $$\prod_{(i,j)\in Cycle} Skew_{ij} = 1$$

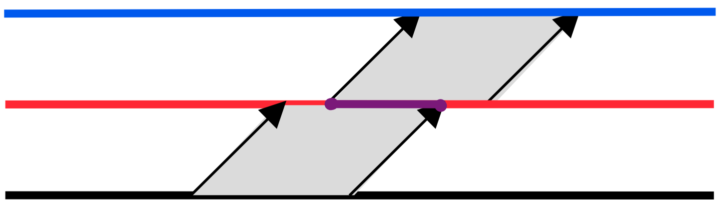


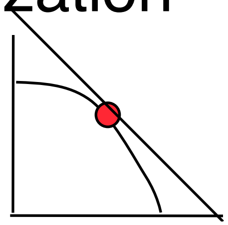
- At this point
  - All MITMs that are not conforming to consistent timing are caught

# Phases of protocol (3)

- Attack on all half-duplex MITMs
  - Every pair of neighboring nodes sends a long packet that exceeds the round-trip delay



- Repeat as needed: Nodes have view of the network
  - Which sets of nodes can concurrently transmit
  - Link-state including clock-synchronization parameters
- Choice of operating point for Network Utility Maximization
  - Optimal network resource scheduling is chosen
  - And agreed to by all nodes



# Phases of protocol (4)

- Data transfer phase
  - The nodes send their data
    - » Over the agreed paths
    - » According to the agreed schedule
    - » Relaying taking place according to the schedule

- Verification of operation
  - Route prefix verification is done to ensure that nodes are conforming
  - Can identify concurrent transmission sets that are not reliable
  - Detected non-conforming concurrent transmission sets are eliminated and network view is established all over again

Thank you