* Real Time Video Streaming in Multi hop Ad-Hoc Wireless networks

Yaniv Fais

Under supervision of Prof. Guy Even

Joint work with Prof. Guy Even, Moti Medina,
Moni Shahar and Alexander Zadorojniy

* Background

- Research part of consortium dedicated to communication for rescue teams
  - Quick deployment (no backbone)
  - Common technology (WiFi)
  - Dedicated system
- Focused on video data
- Practical solution

Agenda

- Problem Definition
  - Real-Time Video
  - WiFi introduction
- Routing & Scheduling
  - Interference models
- Simulator
  - Flow Control
- Experimental results
  - Video transmission

* Problem Definition

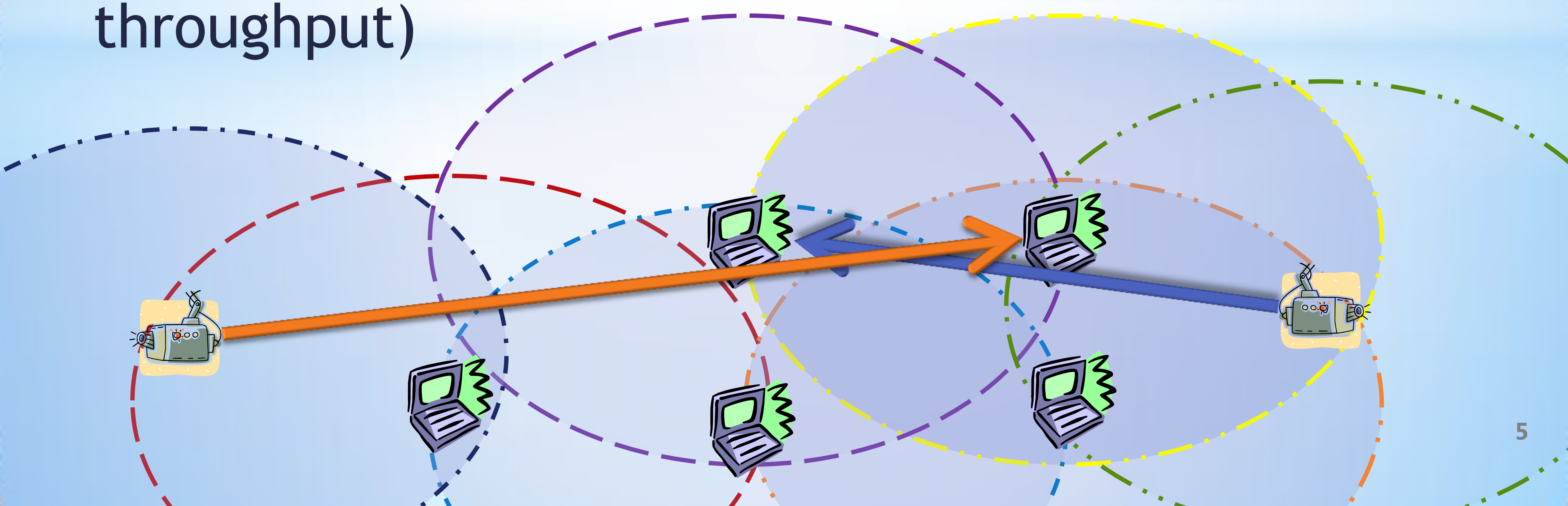
- Ad-Hoc Wireless Network model
  - Nodes in the plane , know location and time (using GPS)
  - Nodes with single radio
  - Multiple radio channels

* Problem Definition II

- Input:
  - Nodes / locations / transmission power
  - real-time video transmission requests:
    - Source , Destination , Bandwidth

(Source and destinations may not be neighbors)

Goal: serve requests fairly (maximize min-throughput)



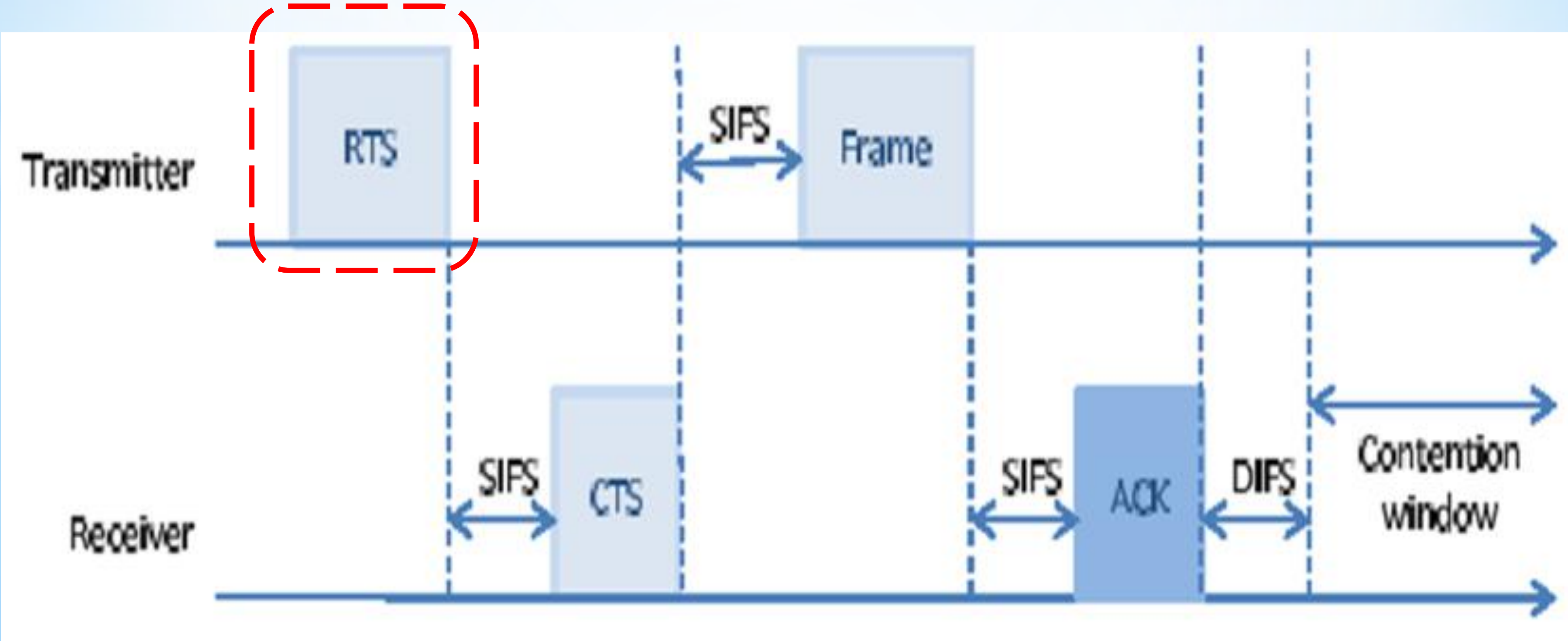
* Real-Time Video characteristics

- High-bandwidth
  - ~1Mbps after encoding
- Require: low end-to-end delay
  - Few seconds at most
- Require: small packet-loss
  - < 5%
- Adjustable bit rate because encoder can be controlled (variable quality and compression ratio)

* IEEE 802.11g (WiFi)

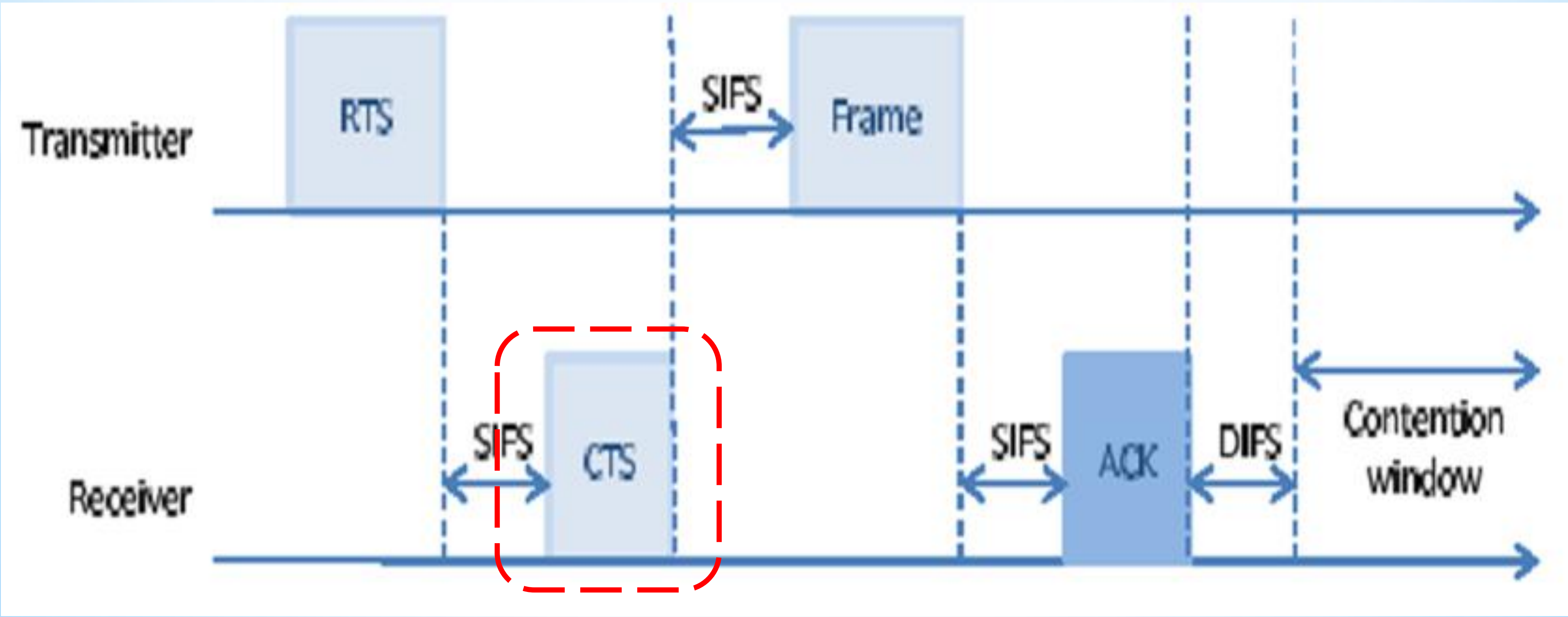
Carrier sense multiple access with collision detection (CSMA/CD)

- Transmitter sends Request To Send (RTS)



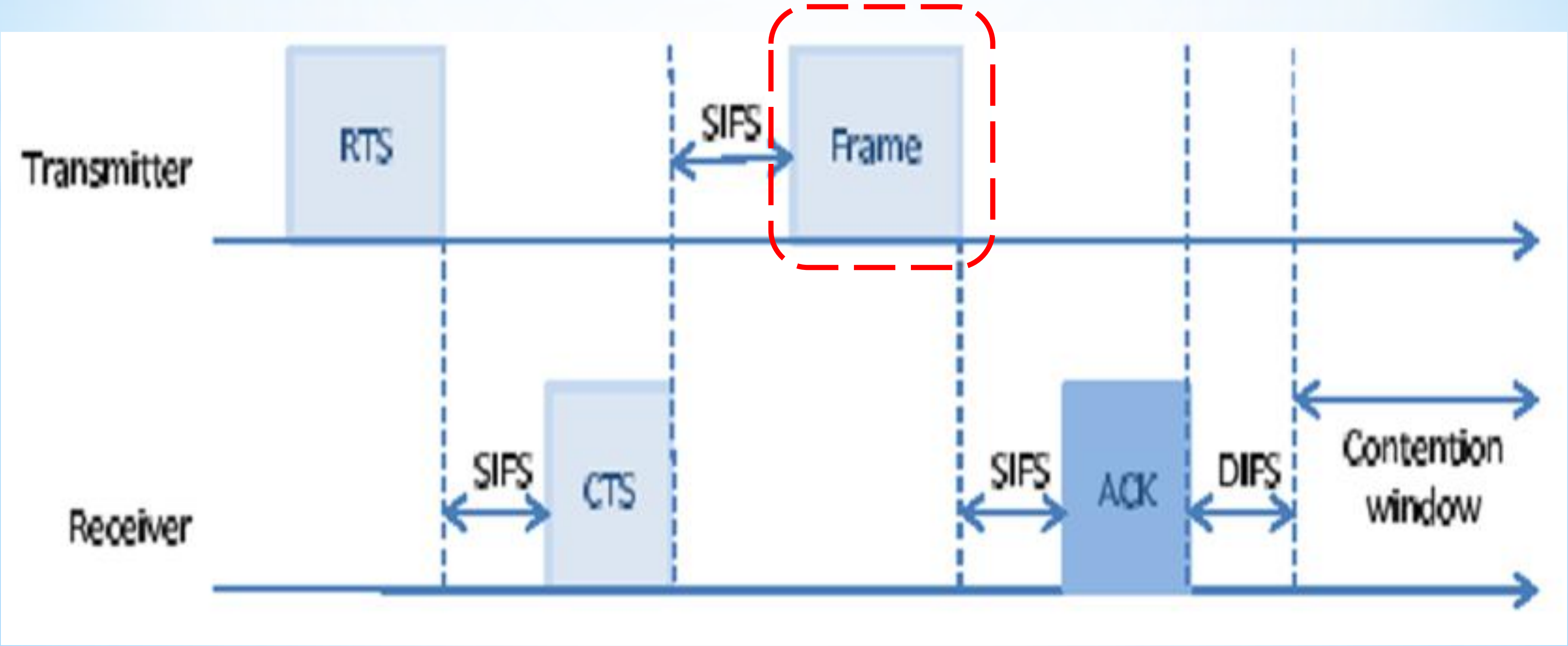
* IEEE 802.11g (WiFi)

- Receiver sends "Clear To Send" (CTS)
- All other nodes remain "quiet" throughout packet transmission



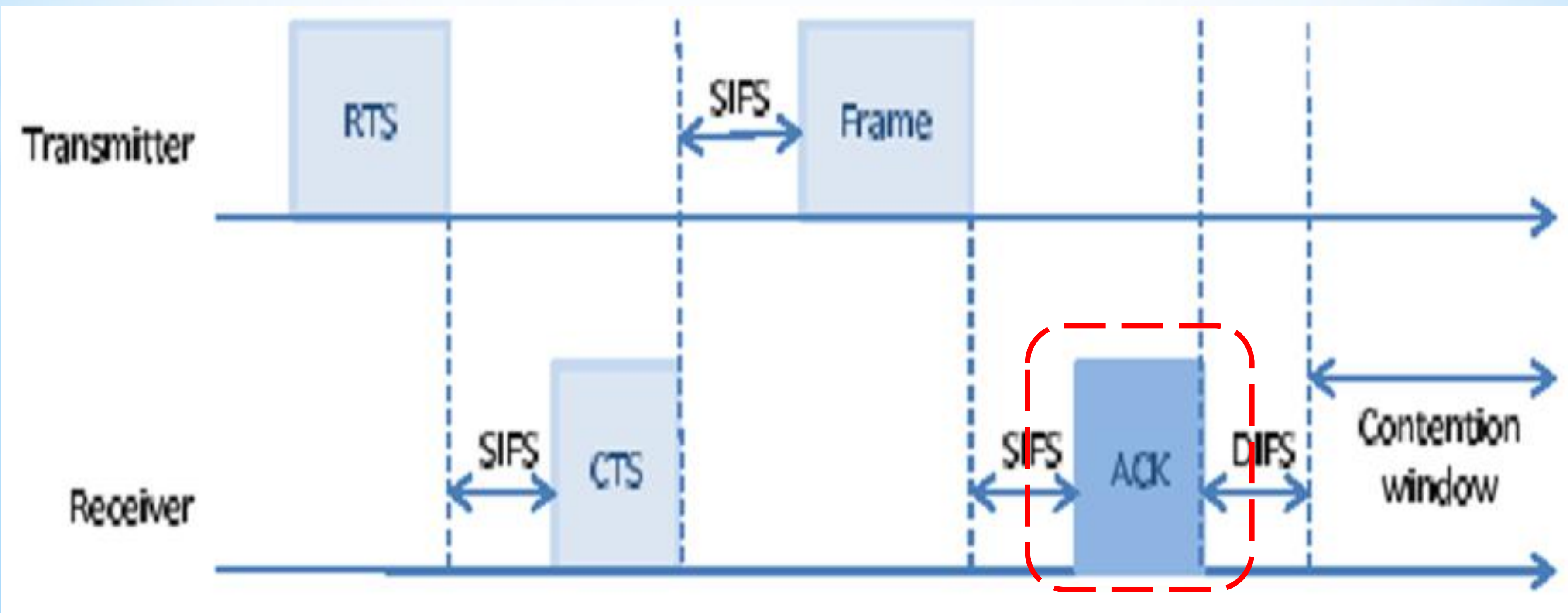
IEEE 802.11g (WiFi)

- Transmitter sends frame with payload data (large frame – several KB)



IEEE 802.11g (WiFi)

- Receiver sends "Acknowledge" (ACK) packet once completed



* IEEE 802.11g (WiFi)

Distributed protocol that avoids collisions

But: (In multi-hop setting)

- Large overhead (for each packet)
- Unpredictable: no quality of service guarantee (throughput, end-to-end delay)
- Unfair (starvation)
- Unstable (unbounded queues)

Research goals

Full system solution for specific video streaming problem:

- Inputs gathering
- Scheduling & Routing computation
- Flow control
- Simulation
- Results analysis

Solution sketch - Time division

- Divide time unit into slots and in each time slot schedule a <u>non-interfering</u> set of links
- Example:
  - <u>Slot 1</u>: Node 1 send to node 22 in <u>channel 2</u>, rate 48Mbps in stream 3
  - <u>Slot 1</u>: Node 10 send to node 2 in <u>channel 2</u>, rate 6Mbps in stream 5
  - Slot 2: ….
- Compute schedule table from node locations in a central node and broadcast to all nodes
- Nodes follow schedule periodically

* Wireless interference model

Signal-to-Interference-and-Noise (SINR) Ratio model

u transmits to *v* while group *S* also transmits:

$$SINR(\mathrm{u,v,S}) = \frac{P/d(u,v)^{\propto}}{N + \sum_{x \in S \setminus u} P/d(x,v)^{\propto}}$$

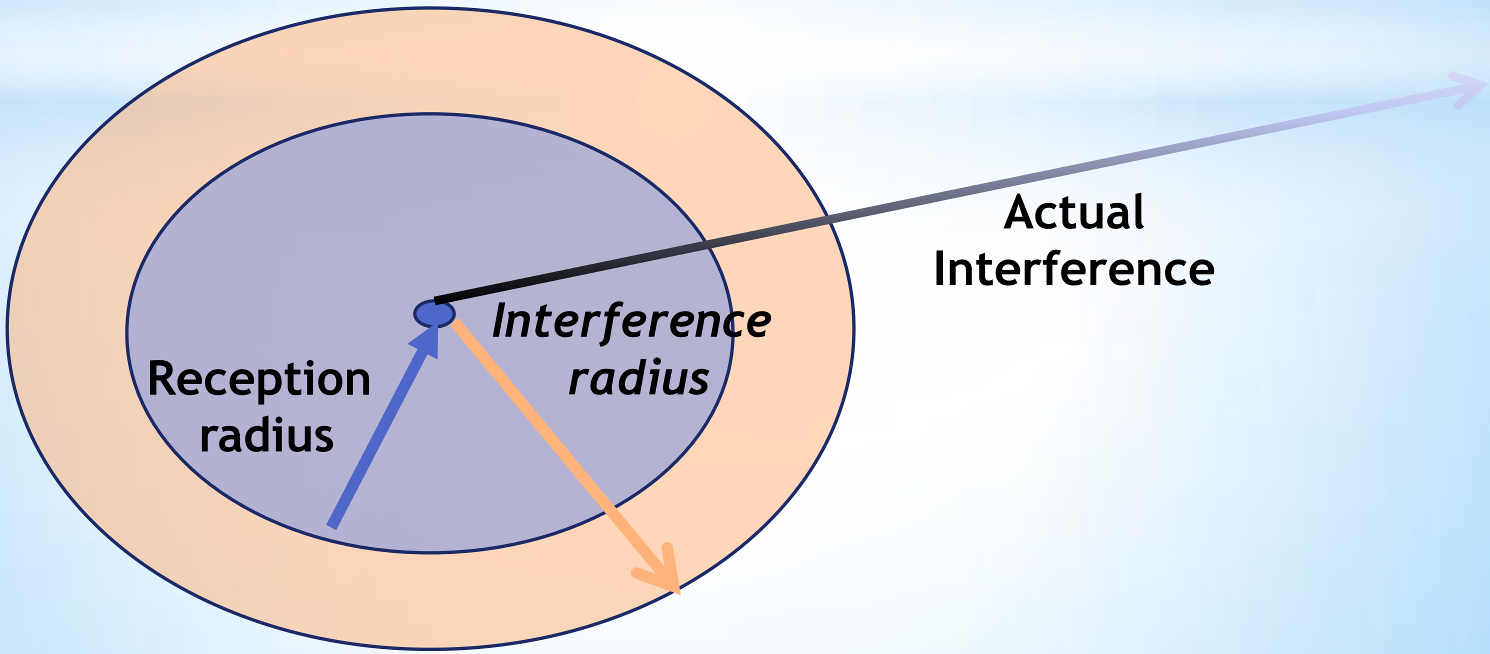
- Message succeeds if SINR(u,v,S) > β_m
- P - transmission power
- *d(u,v)* - distance between node u and v
- N - thermal noise
- $\propto$ - signal decay exponent (usually 2-6)
- β_m – minimum SNR for modulation scheme

Difficult for computation

Wireless interference model

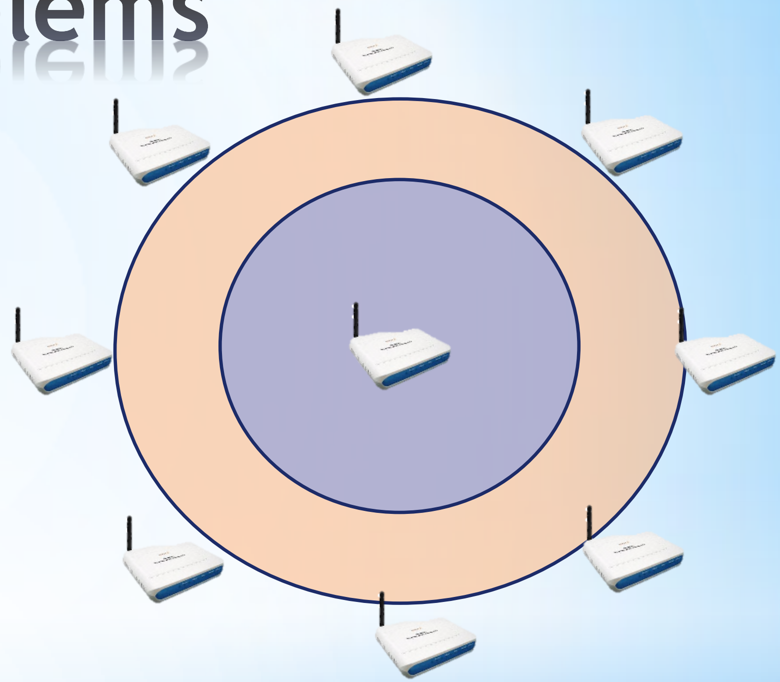
Protocol (Graph) model

- Reception radius – upper bound on distance of transmitter
- Interference radius – reception is successful only if no other transmitter within interference radius

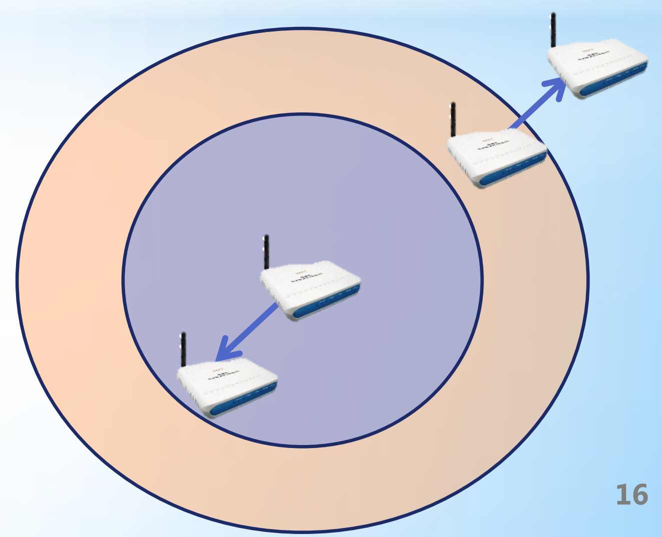


Graph model - problems

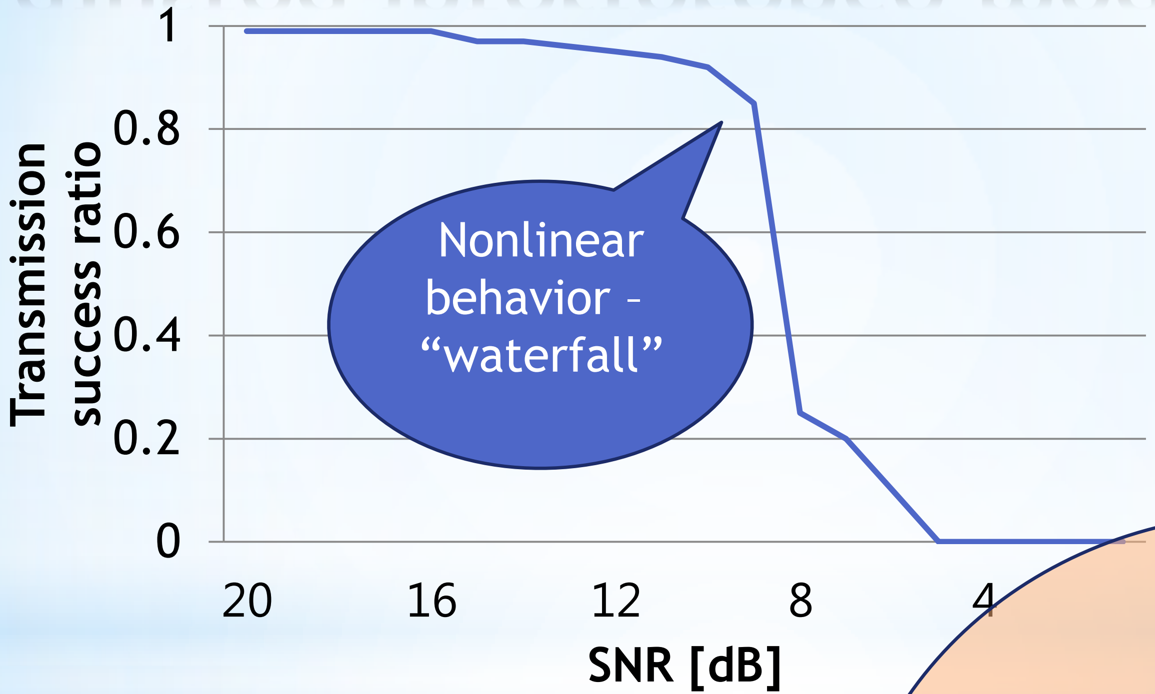
- Optimistic
  - Ignores the additive effect of multiple interferences



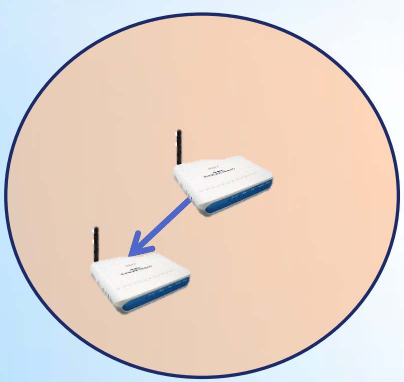
- Pessimistic
  - Excludes concurrent communication of two neighboring links

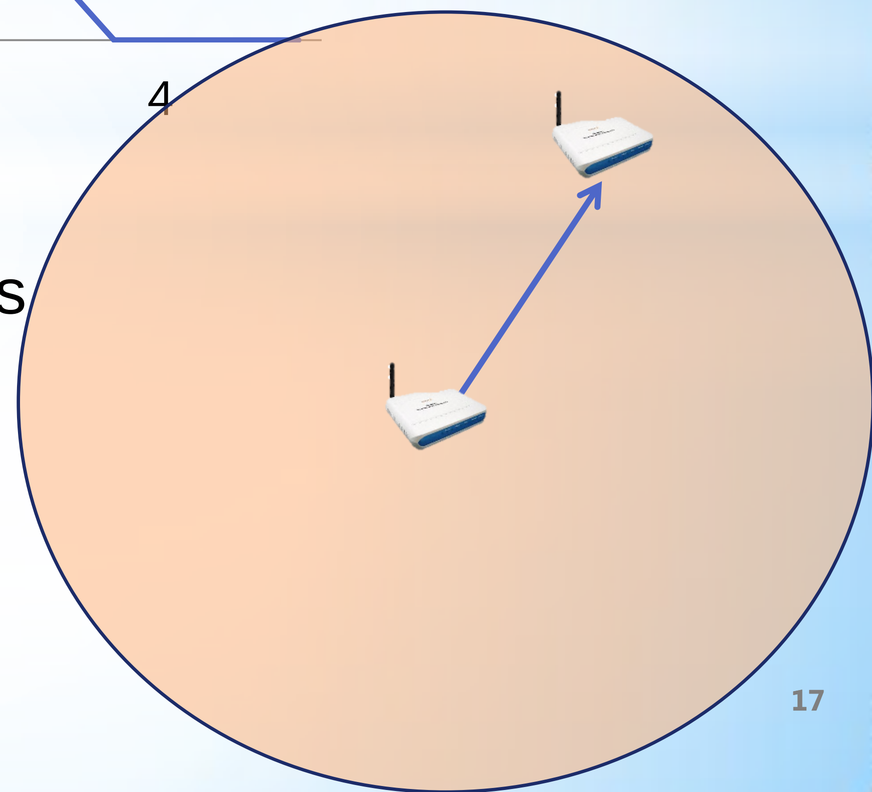


* Adjusted interference model



Interference radius depends on SINR (distance)





17

* WiFi adjusted interference model

- interference set $V_{u,v,m}$ of the link e = (u,v,m)

$V_{u,v,m} \equiv \{x \in V \setminus \{u\} \mid SINR(u, v, \{x\}) < \mu\beta_m$ or $SINR(v, u, \{x\}) < \mu\beta_0\}$

- Interfering set of edge to link *e* :

$I_{u,v,m} \equiv \{e' = (u', v', m') \mid$

$\{u', v'\} \cap (V_{u,v,m} \cup V_{v,u,m}) \neq \emptyset\} \setminus \{(u, v,m)\}$

* Routing & Scheduling

Formalize linear program:

Flow along edge *e*, in channel *f* for stream *i*

$$f_i^j(e) \geq 0 \qquad \forall i \in [1..k], \forall j \in [1..3], \forall e \in E \tag{B.2}$$

$$\sum_{j=1}^{3} f_i^j(e) = f_i(e) \qquad \forall e \in E \tag{B.3}$$

$$\sum_{i=1}^{k} f_i^j(e) = f^j(e) \qquad \forall e \in E$$

Linear Program (cont.)

Flow conservation

$$\sum_{e \in E_{out}(v)} f_i(e) - \sum_{e \in E_{in}(v)} f_i(e) = 0 \qquad \forall i \in [1..k], \forall v \in V \setminus \{a_i, b_i\} \tag{B.4}$$

Capacity constraint

$$\sum_{i=1}^{k} f_i(e) \leq c(e) \qquad \forall e \in E \tag{B.5}$$

Conflict constraint

$$\frac{f^j(e)}{c(e)} + \sum_{j'<j} \sum_{e' \in E(u) \cup E(v)} \frac{f^{j'}(e')}{c(e')} + \sum_{e' \in I_e} \frac{f^j(e')}{c(e')} \leq 1 \qquad \forall e = (u, v, m) \in E, \forall j \in [1..3]$$

- flow/capacity: used time slice

- Only single channel used for transmission
- No transmission done on interfering edges simultaneously

Linear Program (cont.)

Supply ratio for stream i, fraction of supplied demand

$$\sum_{e \in E_{out}(v)} f_i(a_i) - \sum_{e \in E_{in}(v)} f_i(a_i) = d_i^* \cdot \rho_i \qquad \forall i \in [1..k]$$

Demand for stream i

$$\forall i \in [1..k]$$

$$\rho \leq \rho_i \qquad \forall i \in [1..k]$$

- Solve LP for maximum ρ

Routing & Scheduling algorithm

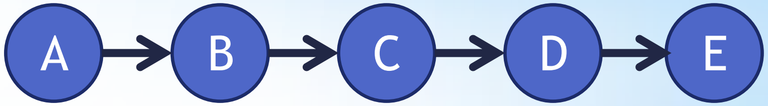
- Formulate linear program (LP) for max-min throughput (graph model).
- Output:
  - $flow\ [packets\ per\ period] - f_i^j(e)$

Need to compute scheduling table that supports the flow. (rounding problem)

Greedy scheduler

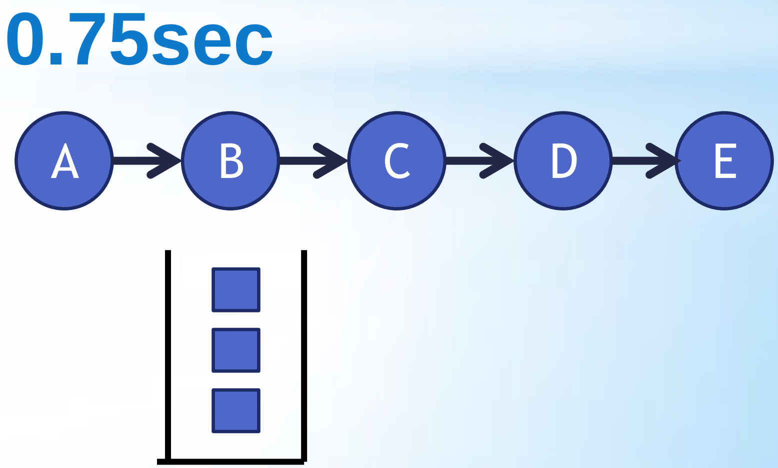
- Greedy coloring of paths
  - Scan the links and frequency channels
  - Assign slots to each link and frequency channel

- Example problem: (Period time: 1 sec.)
  - Slot 4: A -> B
  - Slot 3: B -> C
  - Slot 2: C -> D
  - Slot 1: D -> E



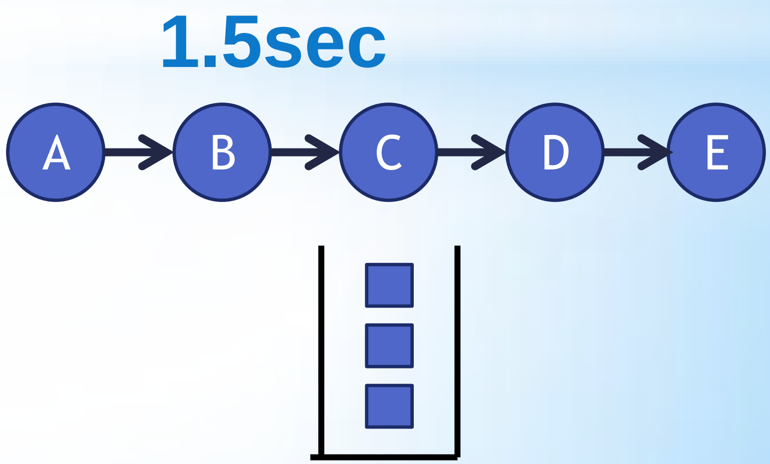
Greedy scheduler

- Example problem: (Period time: 1 sec.)
  - Slot 4: A -> B
  - Slot 3: B -> C
  - Slot 2: C -> D
  - Slot 1: D -> E



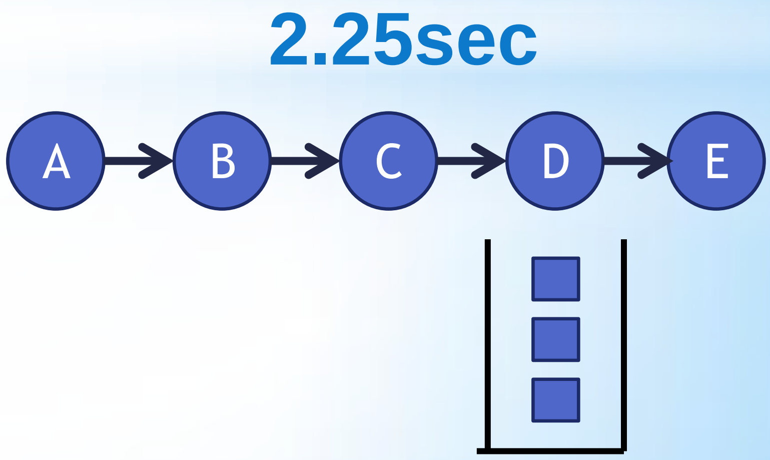
Greedy scheduler

- Example problem: (Period time: 1 sec.)
  - Slot 4: A -> B
  - Slot 3: B -> C
  - Slot 2: C -> D
  - Slot 1: D -> E



Greedy scheduler

- Example problem: (Period time: 1 sec.)
  - Slot 4: A -> B
  - Slot 3: B -> C
  - Slot 2: C -> D
  - Slot 1: D -> E

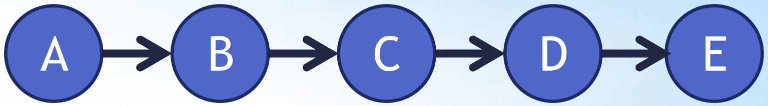


* Greedy scheduler

- Example problem: (Period time: 1 sec.)
  - Slot 4: A -> B
  - Slot 3: B -> C
  - Slot 2: C -> D
  - Slot 1: D -> E

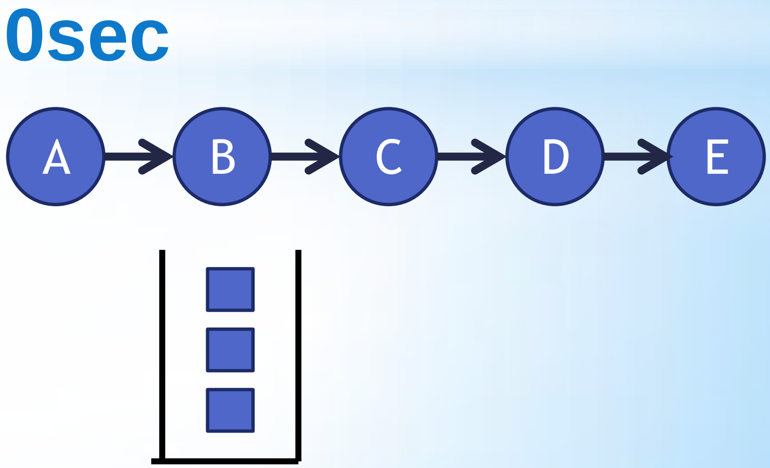
Large latency

3sec



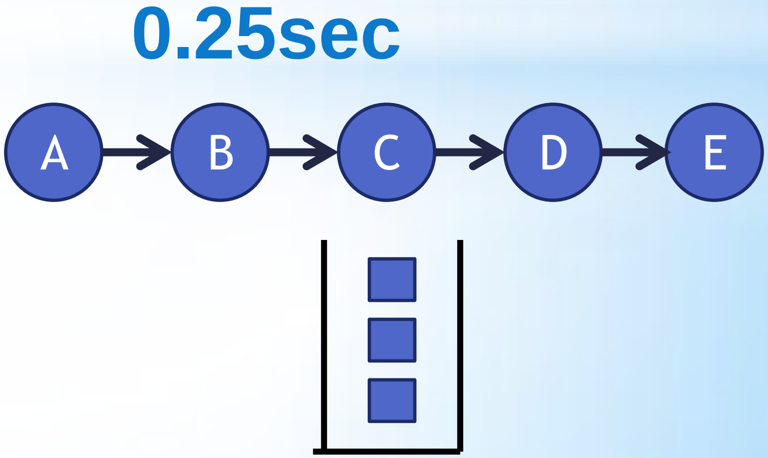
* Better schedule

- Example test case: (Period time: 1 sec.)
  - Slot 1: A -> B
  - Slot 2: B -> C
  - Slot 3: C -> D
  - Slot 4: D -> E



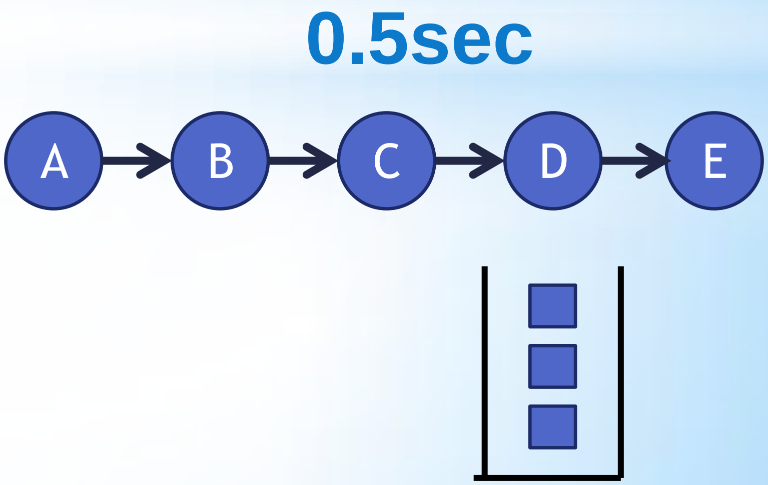
Better schedule

- Example test case: (Period time: 1 sec.)
  - Slot 1: A -> B
  - **Slot 2: B -> C**
  - Slot 3: C -> D
  - Slot 4: D -> E



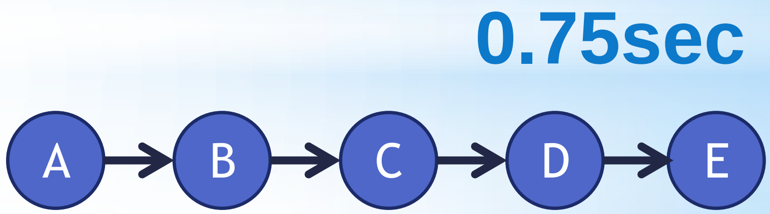
Better schedule

- Example test case: (Period time: 1 sec.)
  - Slot 1: A -> B
  - Slot 2: B -> C
  - **Slot 3: C -> D**
  - Slot 4: D -> E



* Better schedule

- Example test case: (Period time: 1 sec.)
  - Slot 1: A -> B
  - Slot 2: B -> C
  - Slot 3: C -> D
  - **Slot 4: D -> E**



Path-peeling scheduler

- Decomposes each flow into flow paths such that the flow along each path **equals the bottleneck**, i.e., the minimum packets-per-slot(e) along the path. $Let \ \{f_i(p)\}_{p \in P(i)}$ denote the decomposition.

- While not all the flow is scheduled:
  - For i = 1 to k do:
  - If P(i) ≠ ∅ then schedule a path p ∈ P(i) and remove p from P(i).

Routing & Scheduling

- Approximately computed in graph model.

Does the result work in the physical (SINR) model ?

* Simulator

OMNET++/MiXiM – event driven network simulator

- Extensible (C++)
- Built-In wireless protocols (802.11b-Mac / Phy)
- Wireless signal model (noise , interference , fading)
- Mobility
- Fast - Models packets (not symbols)

System stability

Successful delivery is a random variable.

- white noise in channel
- 802.11 random quiet periods

Fluctuations in number of packets sent in each slot incur instable queue lengths

Flow control algorithm

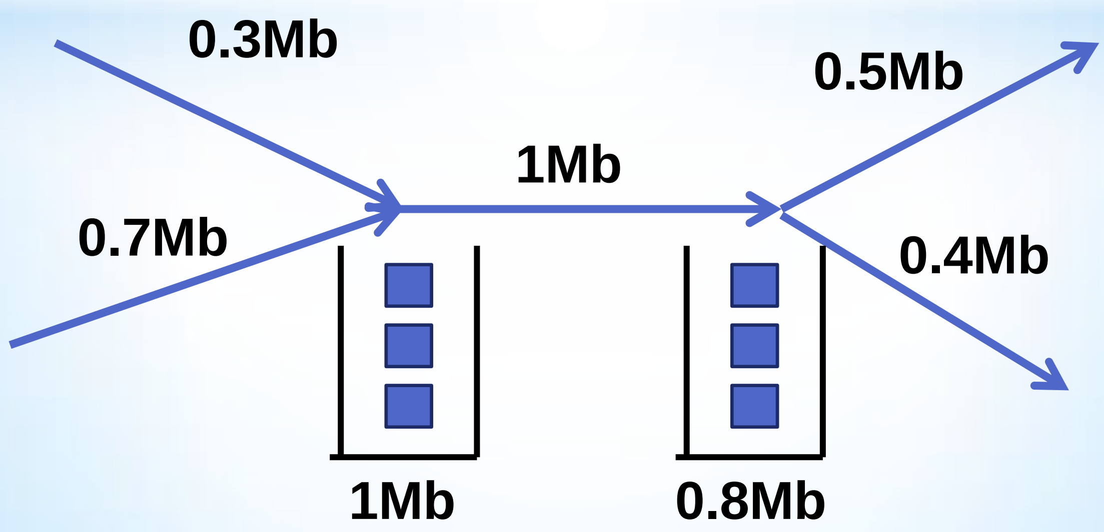
Goal: stabilize queue size.

* Flow Control algorithm

- At the end of each schedule period each node locally performs (per stream):
  - Counts #packets transmitted along outgoing & incoming edges.
  - Sends updated requests for incoming link rates.
  - Drops the oldest packets from queue so that all pending packets are delivered within the next period.

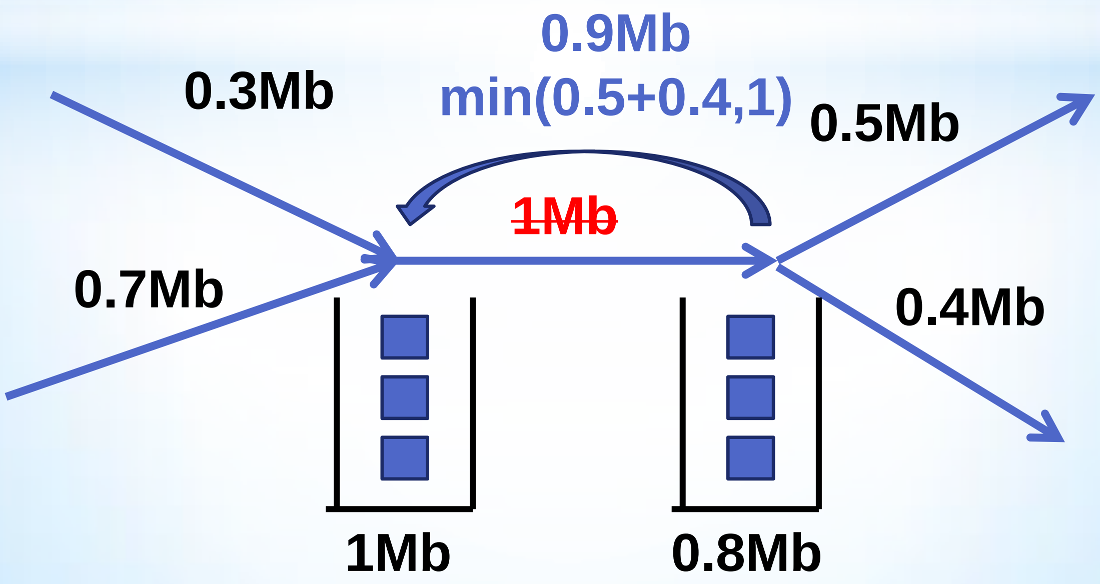
* Flow Control algorithm example

- End of schedule period
- Values indicate data transmitted across edges during last period (per stream)



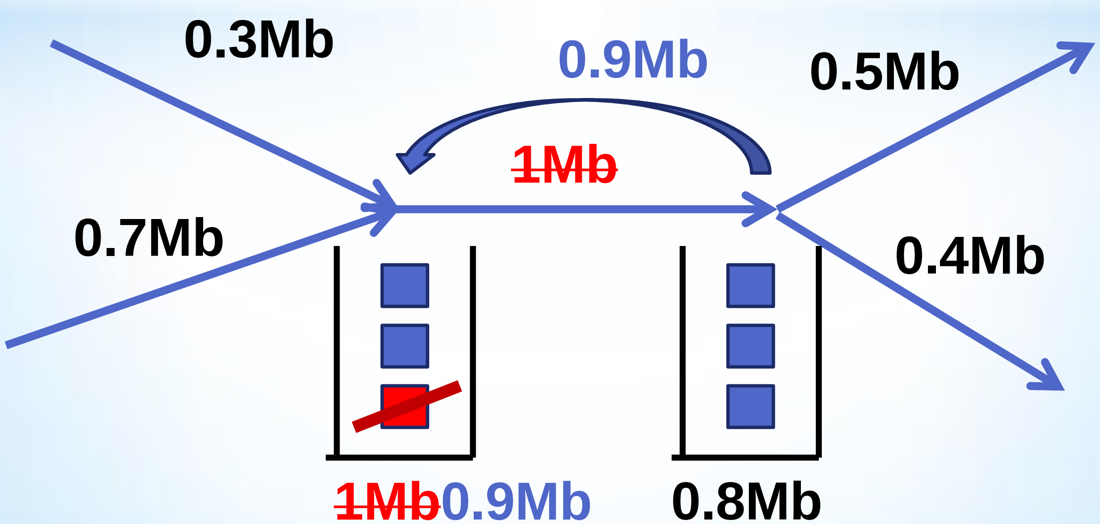
Flow Control algorithm example

- End of schedule period
- Values indicate data transmitted across edges during last period (per stream)
- Update step #1:



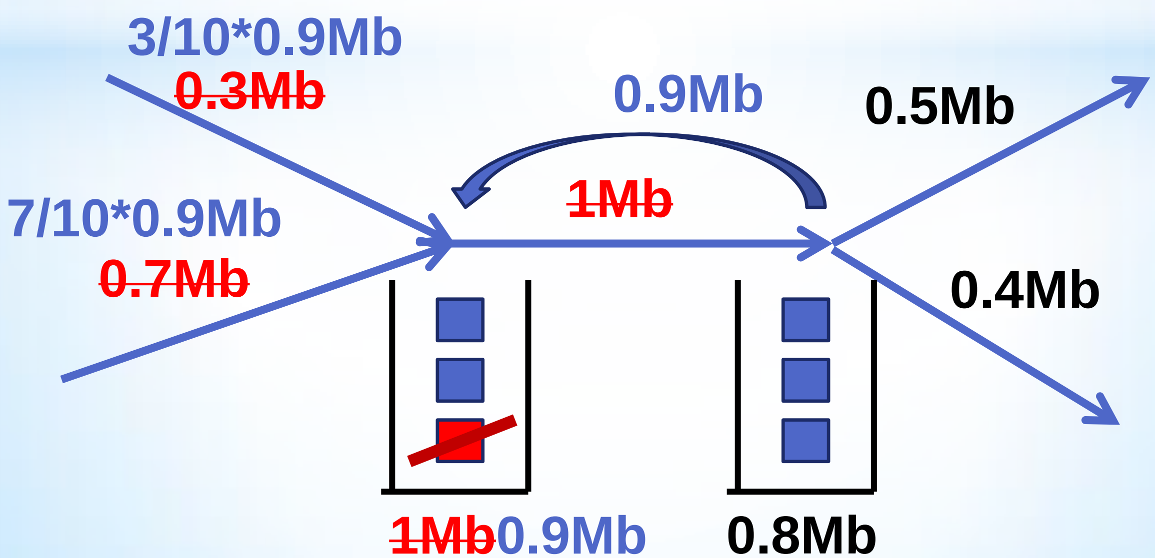
Flow Control algorithm example

- End of schedule period
- Values indicate data transmitted across edges during last period (per stream)
- Update step #2:

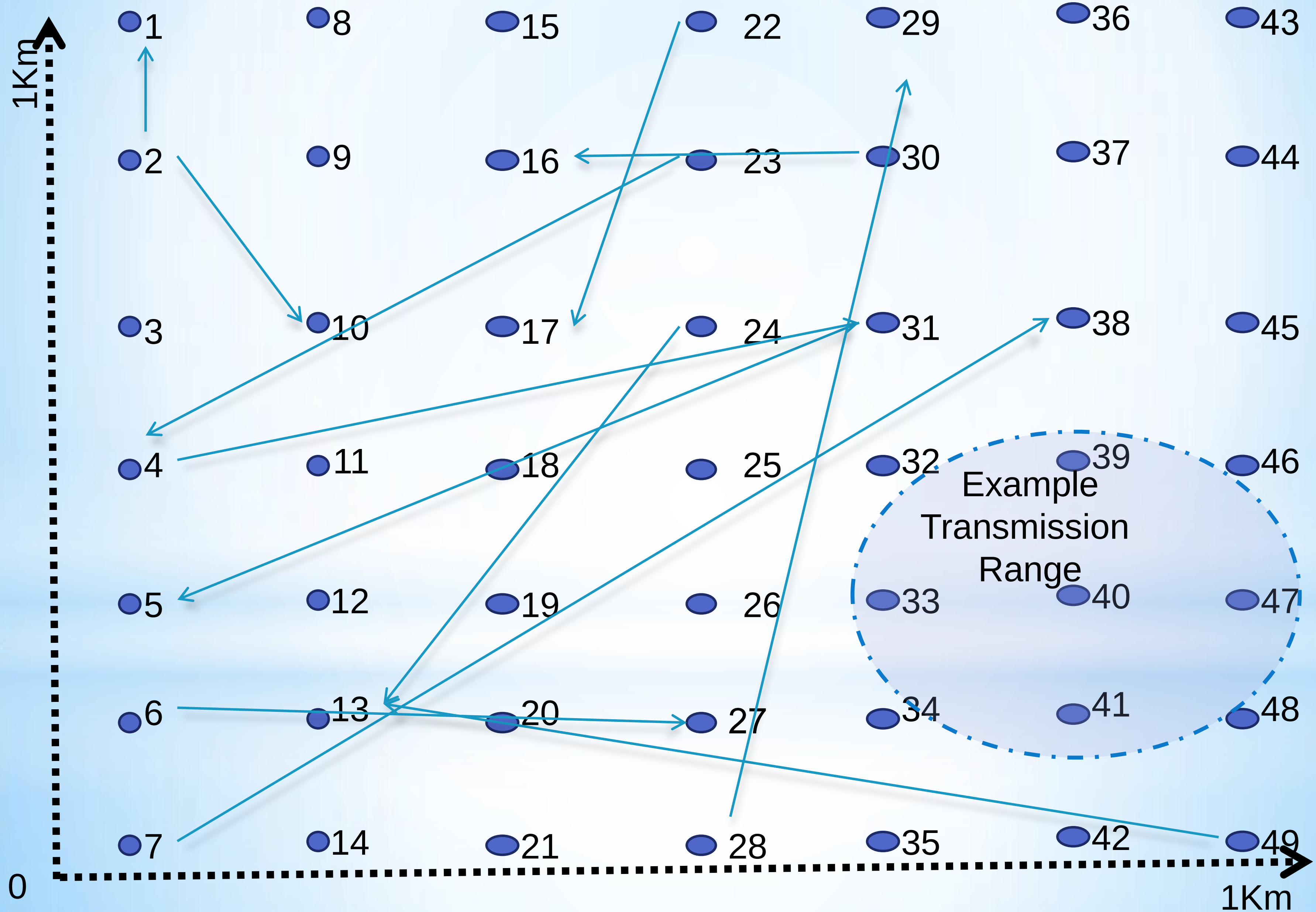


Flow Control algorithm example

- End of schedule period
- Values indicate data transmitted across edges during last period (per stream)
- Update step #3:

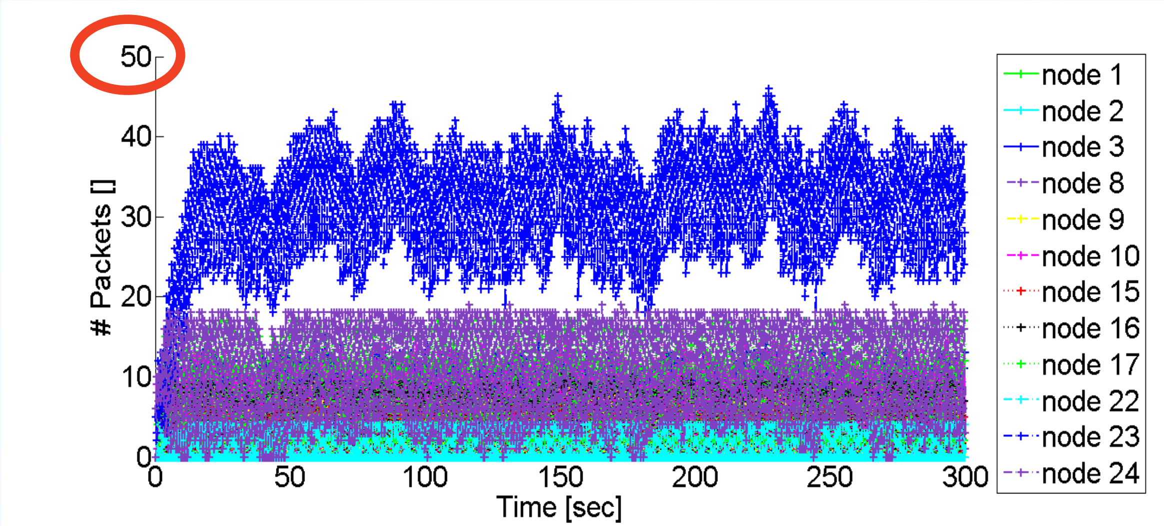


Example Input scenario – Nodes spaced equally on grid with random requests

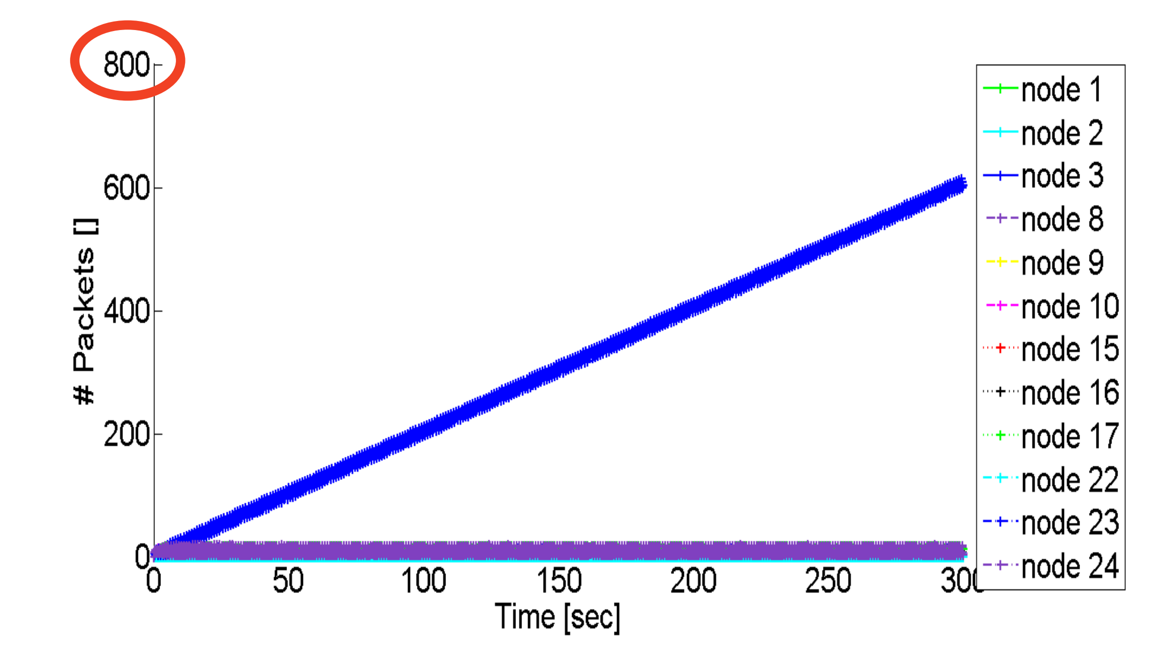


Queue size (one stream/different nodes)

With flow control

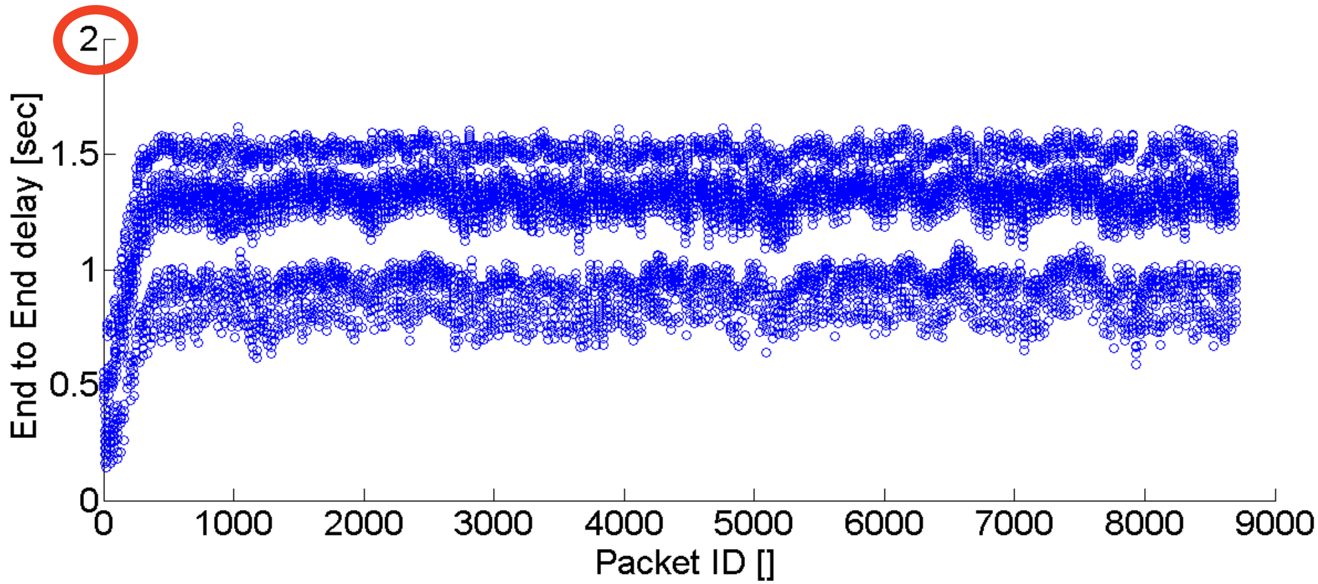


Without flow control

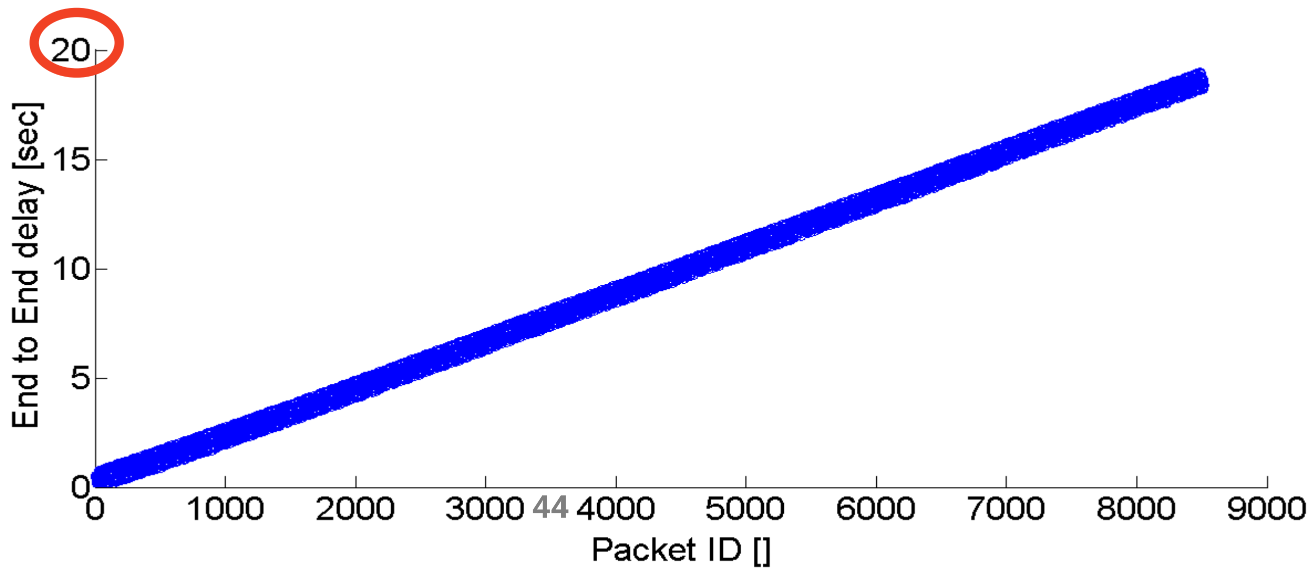


End-to-end delay (one stream)

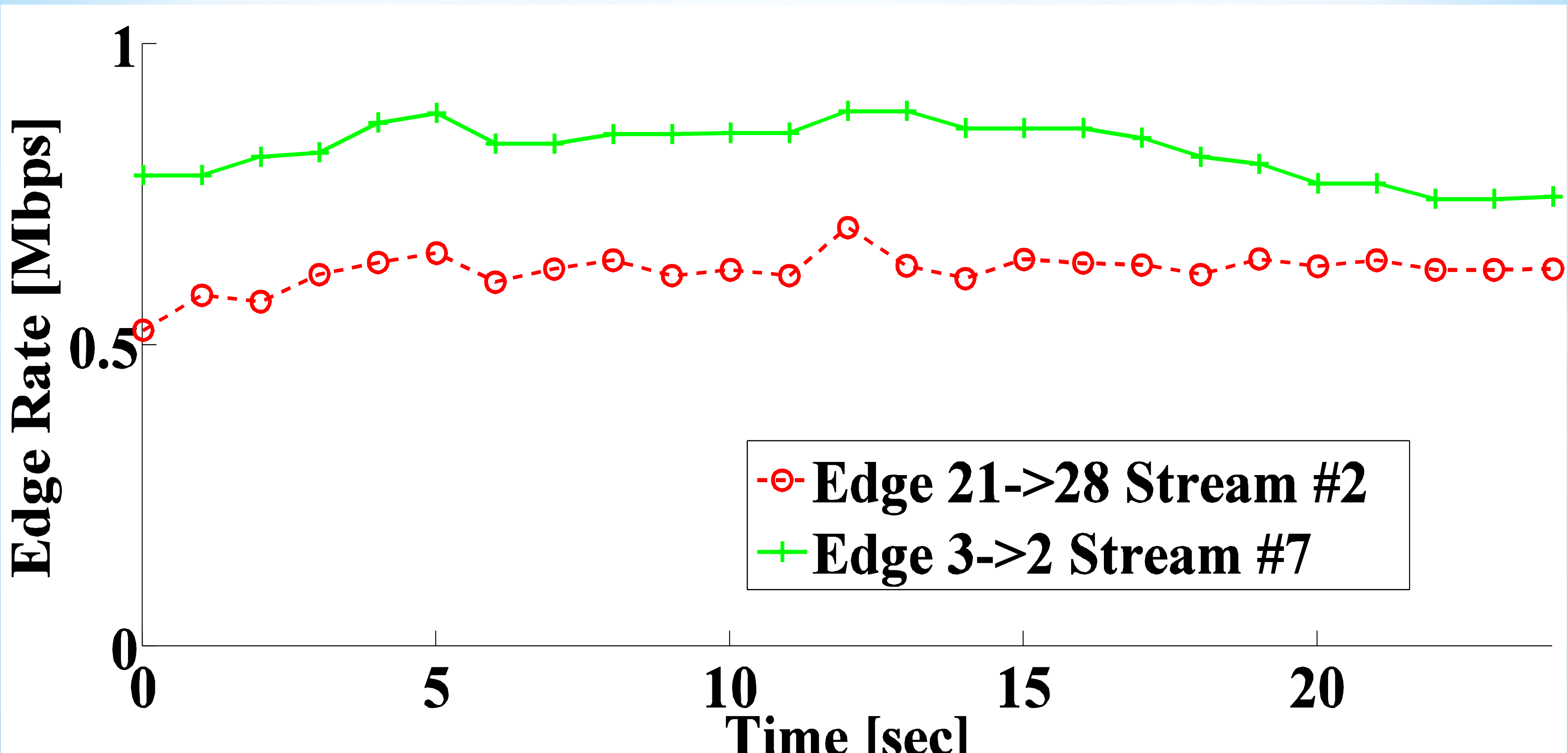
With flow control



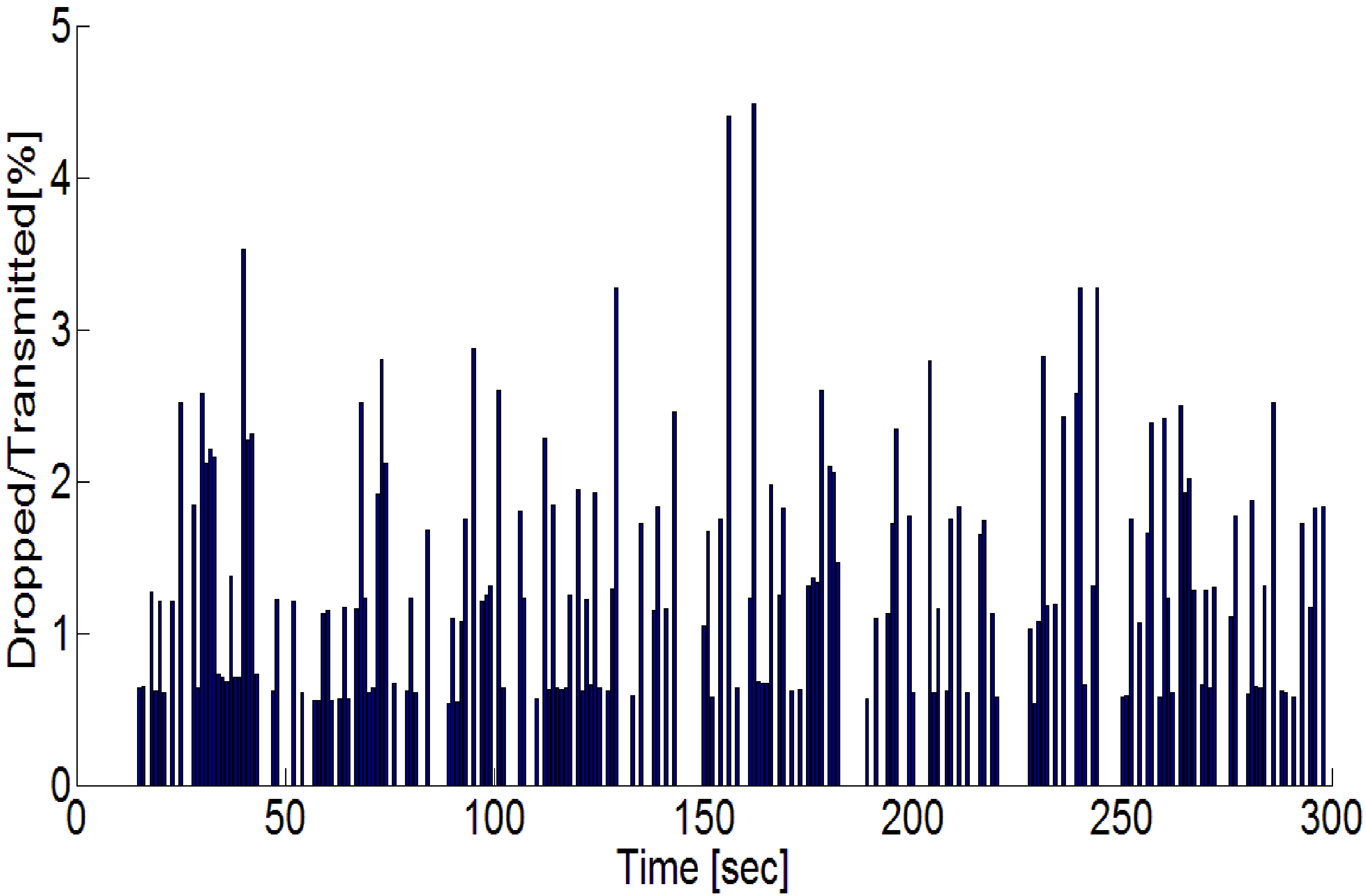
Without flow control



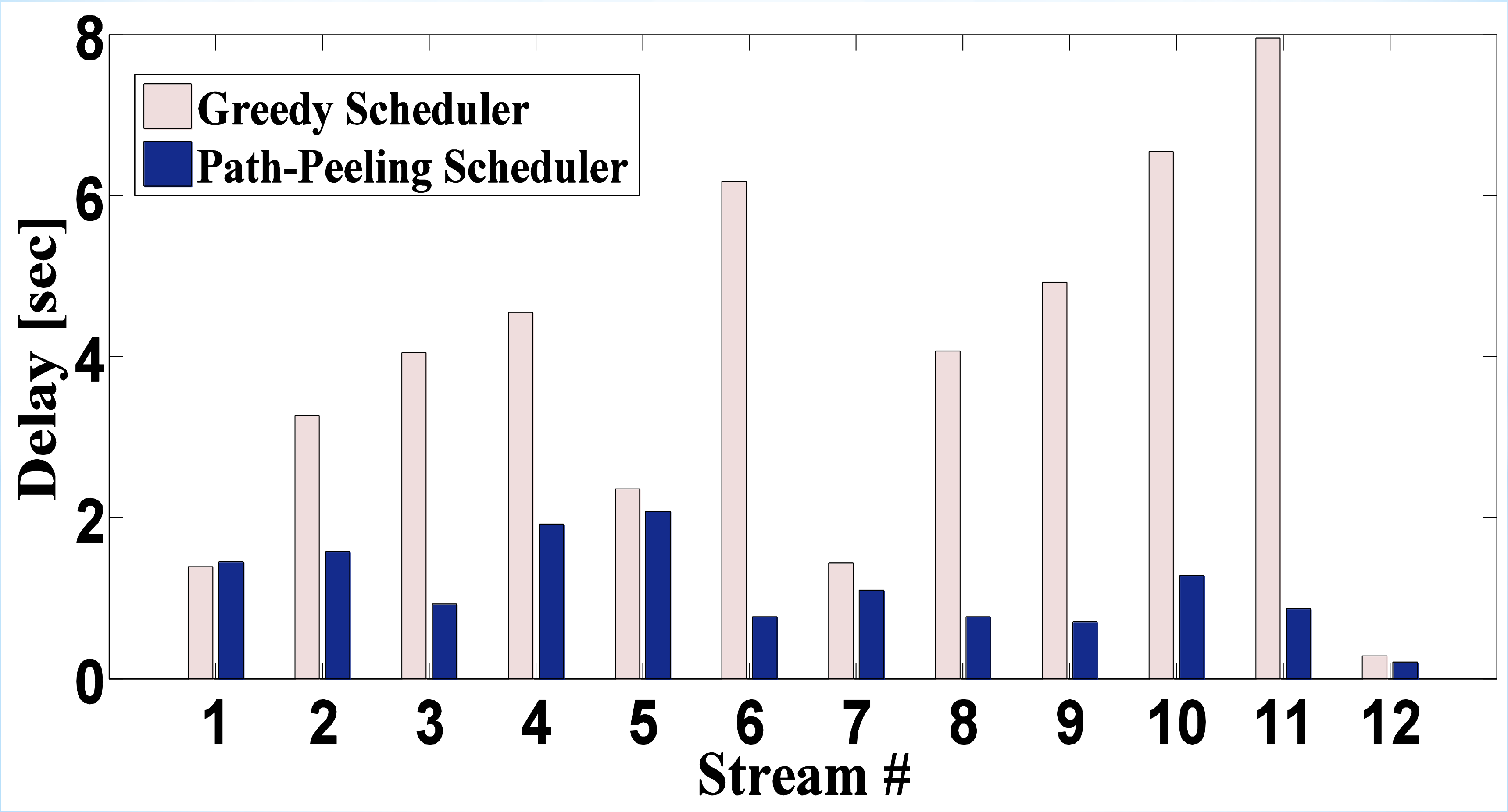
* Flow-control rate changes over time



*Drop rate (worst stream)



End-to-End delay (per scheduler)



47

* Summary

- Routing & scheduling approximation algorithm
- Feasible in physical model (by simulation)
- Local flow control algorithm
- Support real-time video streaming requirements

*Questions ?

*Backup slides

*Video playback example

* WiFi adjusted interference model

- interference set $V_{u,v,m}$ of the link e = (u,v,m)

  $V_{u,v,m} \equiv \{x \in V \setminus \{u\} \mid SINR(u, v, \{x\}) < \mu\beta_m$ or $SINR(v, u, \{x\}) < \mu\beta_0\}$

- Interfering set of edge to link *e* :

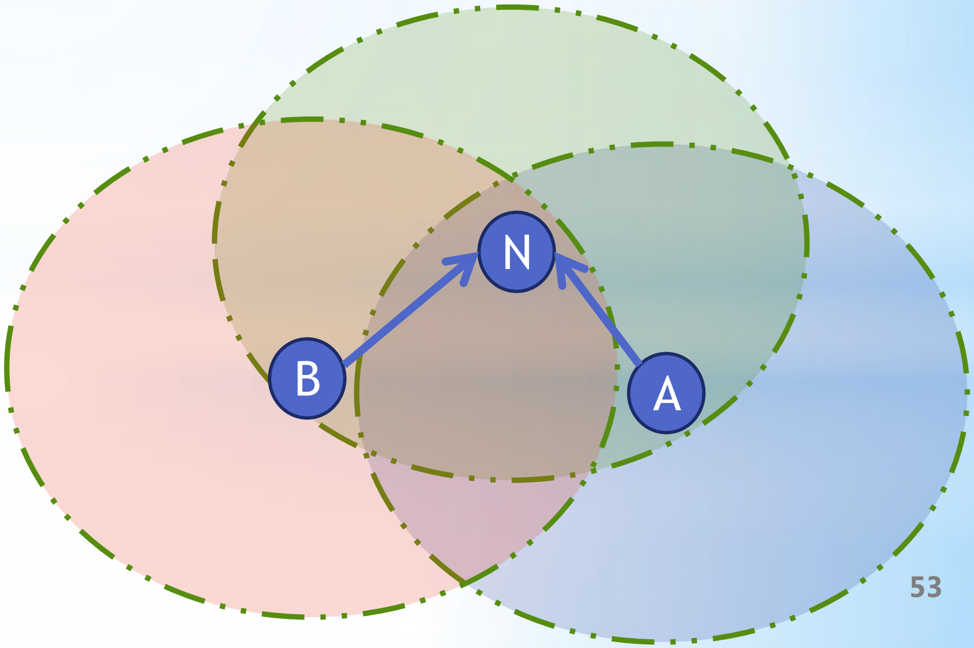
  $I_{u,v,m} \equiv \{e' = (u', v', m') \mid$

  $\{u', v'\} \cap (V_{u,v,m} \cup V_{u,v,m}) \neq \emptyset\} \setminus \{(u, v, m)\}$

Hidden node problem

- A and B can communicate with N
- A can’t “hear” B
- B can’t hear A

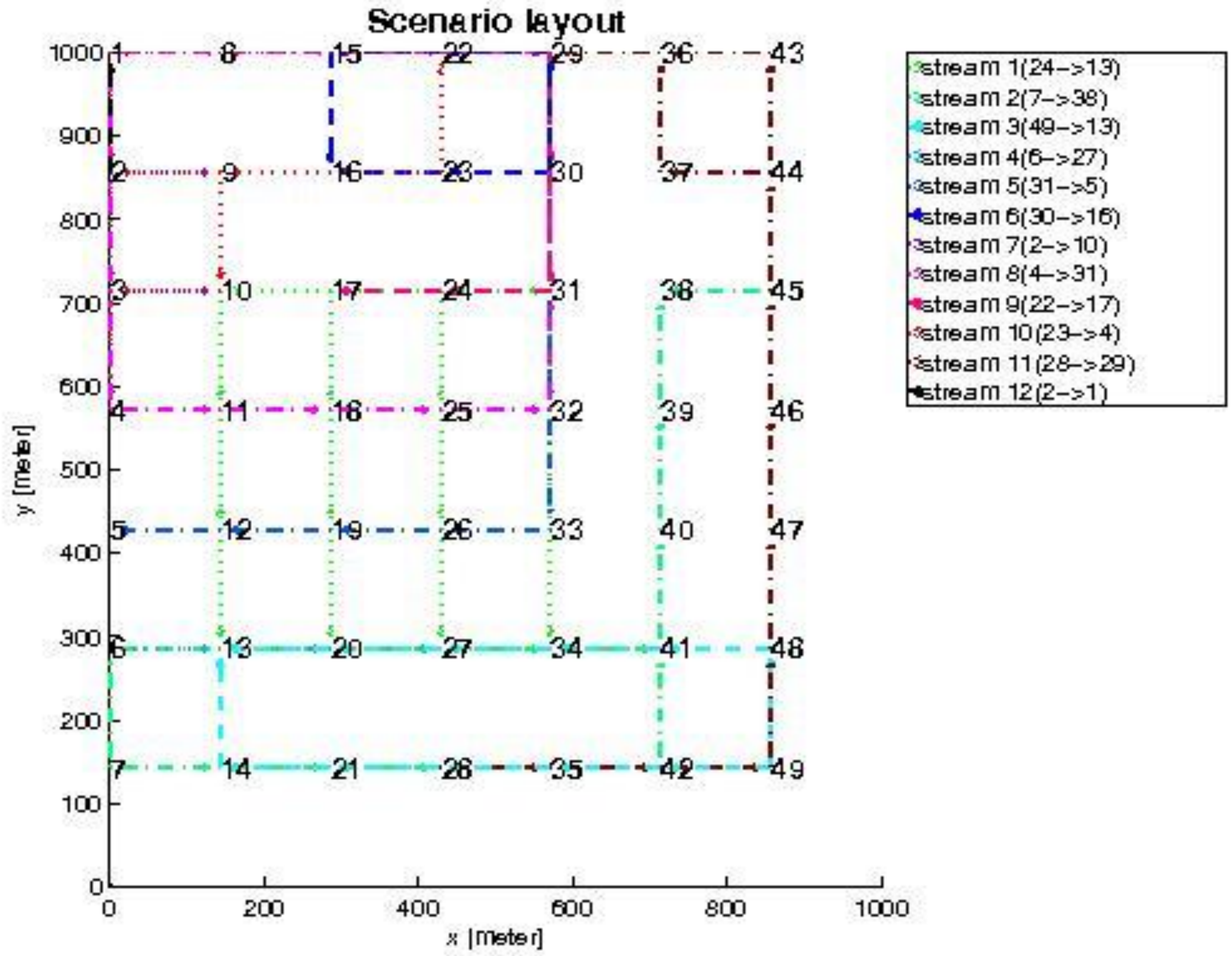
- A and B May attempt to transmit un the same time and interfere with each other!



Routing

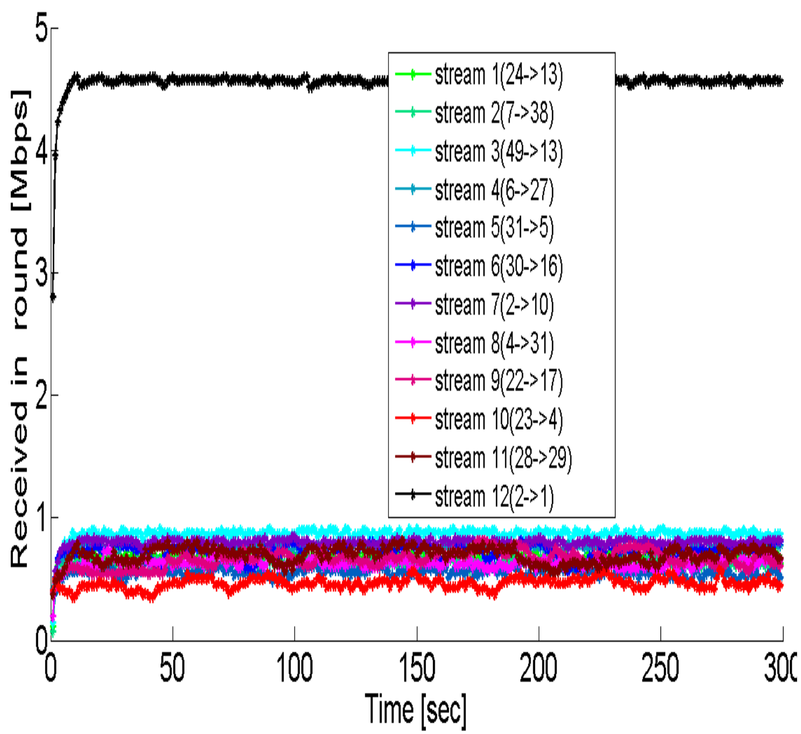
- NP Hard problem in general
- Formalize constraints:
  - Flow in equals flow out
  - Radio is not used simultaneously for transmit and receive and not used in more than one channel
  - Flow is smaller or equal edge capacity
  - No conflict between transmitting nodes - by interfering edges induces by graph interference model (adjusted for WiFi- both ends transmit and multiple MCS)

Routing result

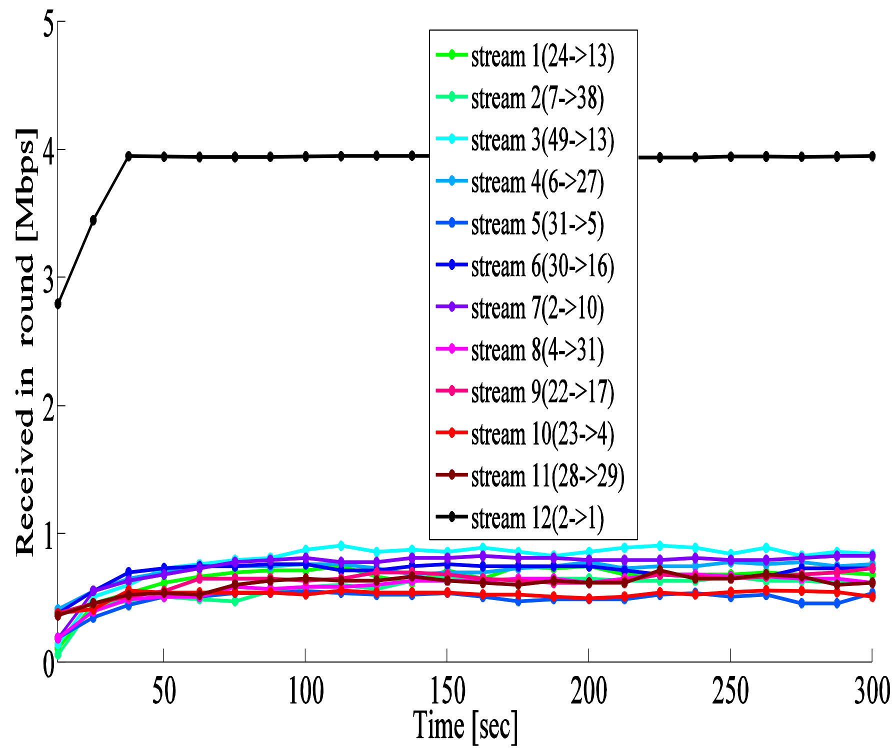


Throughput per stream

With flow control



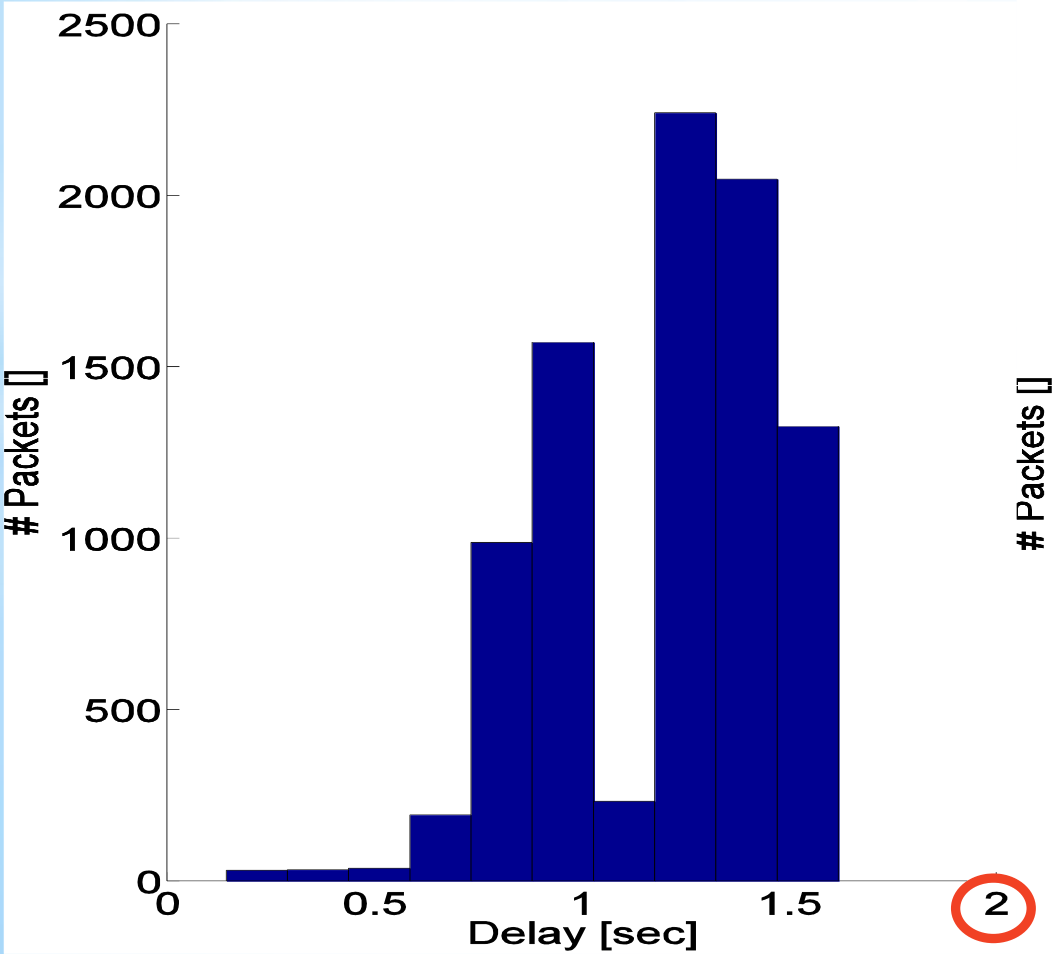
Without flow control



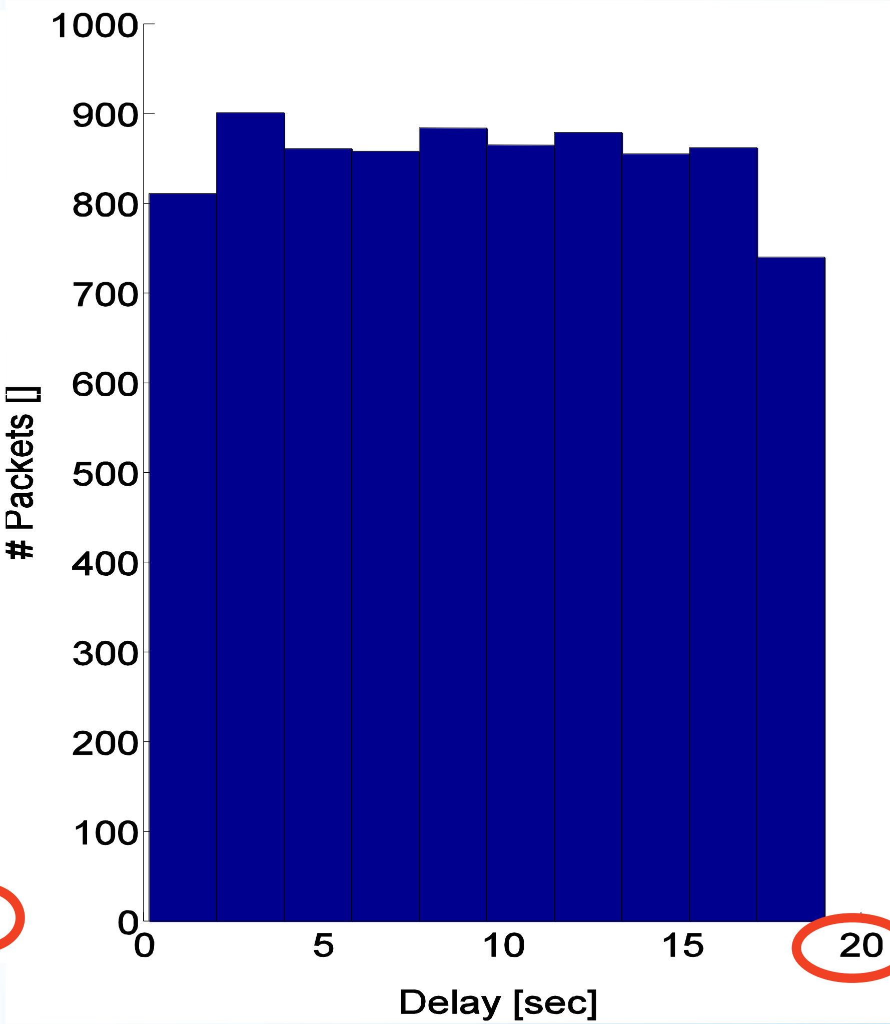
- May even increase ratio

* End-to-end delay histogram (one stream)

With flow control



Without flow control



Routing & Scheduling

Formalize linear program:

2'nd target constant

Demand for stream i

Supply ratio for stream i, fraction of supplied demand

min $_i$ P_i

$$\max \quad \rho + \lambda \cdot \sum_{i=1}^{k} d_i^* \cdot \rho_i \qquad \text{subject to} \qquad (B.1)$$

Flow along edge *e*, in channel *f* for stream *i*

$$f_i^j(e) \geq 0 \qquad \forall i \in [1..k], \forall j \in [1..3], \forall e \in E \qquad (B.2)$$

$$\sum_{j=1}^{3} f_i^j(e) = f_i(e) \qquad \forall e \in E \qquad (B.3)$$

$$\sum_{i=1}^{k} f_i^j(e) = f^j(e) \qquad \forall e \in E$$

* Linear Program (cont.)

Flow conservation

$$\sum_{e \in E_{out}(v)} f_i(e) - \sum_{e \in E_{in}(v)} f_i(e) = 0 \qquad \forall i \in [1..k], \forall v \in V \setminus \{a_i, b_i\} \tag{B.4}$$

Capacity constraint

$$\sum_{i=1}^{k} f_i(e) \le c(e) \qquad \forall e \in E \tag{B.5}$$

$$\sum_{e \in E_{out}(v)} f_i(a_i) - \sum_{e \in E_{in}(v)} f_i(a_i) = d_i^* \cdot \rho_i \qquad \forall i \in [1..k] \tag{B.6}$$

$$\rho \le \rho_i \qquad \forall i \in [1..k] \tag{B.7}$$

Conflict constraint

$$\frac{f^j(e)}{c(e)} + \sum_{j'<j} \sum_{e' \in E(u) \cup E(v)} \frac{f^{j'}(e')}{c(e')} + \sum_{e' \in I_e} \frac{f^j(e')}{c(e')} \le 1 \qquad \forall e = (u, v, m) \in E, \forall j \in [1..3] \tag{B.8}$$

* Flow Control algorithm - definitions

- $P(e,s,t) \equiv$ the number of packets belonging to stream *s* sent along the link *e* during the period *t*
- $P^{+}(e,s,t) \equiv$ the maximum number of packets belonging to stream *s* that can be sent along the link *e* during the period *t*
- $P^{+}(e, s, t) > P(e, s, t)$ iff the relevant queue is empty when a packet is scheduled, if no schedule for *s* on *e* then $P^{+}(e, s, t)=0$

* Flow Control algorithm prelim.

- Algorithm is executed locally for all nodes
- Each node has a queue per stream
- *R(e, s)* specifies the number of packets from stream *s* that the node is willing to receive along the link *e* in the next period *t + 1*
- Initialize: for all $e \in E_{in}(v)$

$R(e, s) \leftarrow$ *[scheduler computed rate]*

Flow Control algorithm

For t = 1 to ∞ do

- Measure $P(e, s, t)$ for every $e \in E(v)$, and $P+(e, s, t)$ for every $e \in E_{out}(v)$.
- Receive $P^{+}(e, s, t)$ for every $e \in E_{in}(v)$, and $R(e, s)$ for every $e \in E_{out}(v)$.
- $Rin \leftarrow \min\{\sum e \in E_{out}(v)\ R(e, s), \sum e \in E_{out}(v)\ P+(e, s, t), \sum e \in E_{in}(v)\ P^{+}(e, s, t)\}$.
- $for\ every\ e \in E_{in}(v)$:

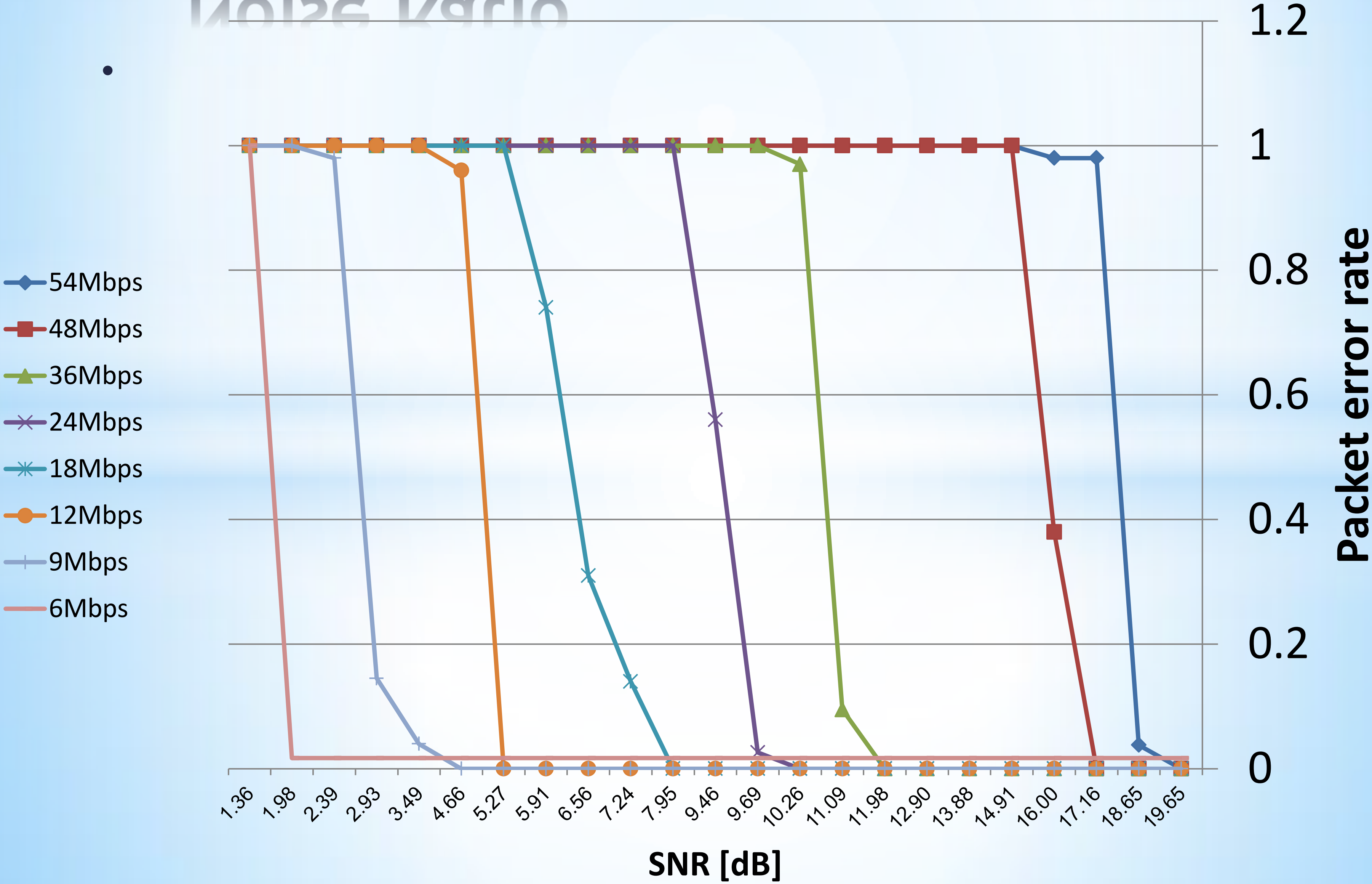
$$R(e, s) \leftarrow Rin * \frac{P^{+}(e, s, t)}{\sum_{e` \in E_{in}(v)} P^{+}(e`, s, t)}$$

* Flow Control algorithm

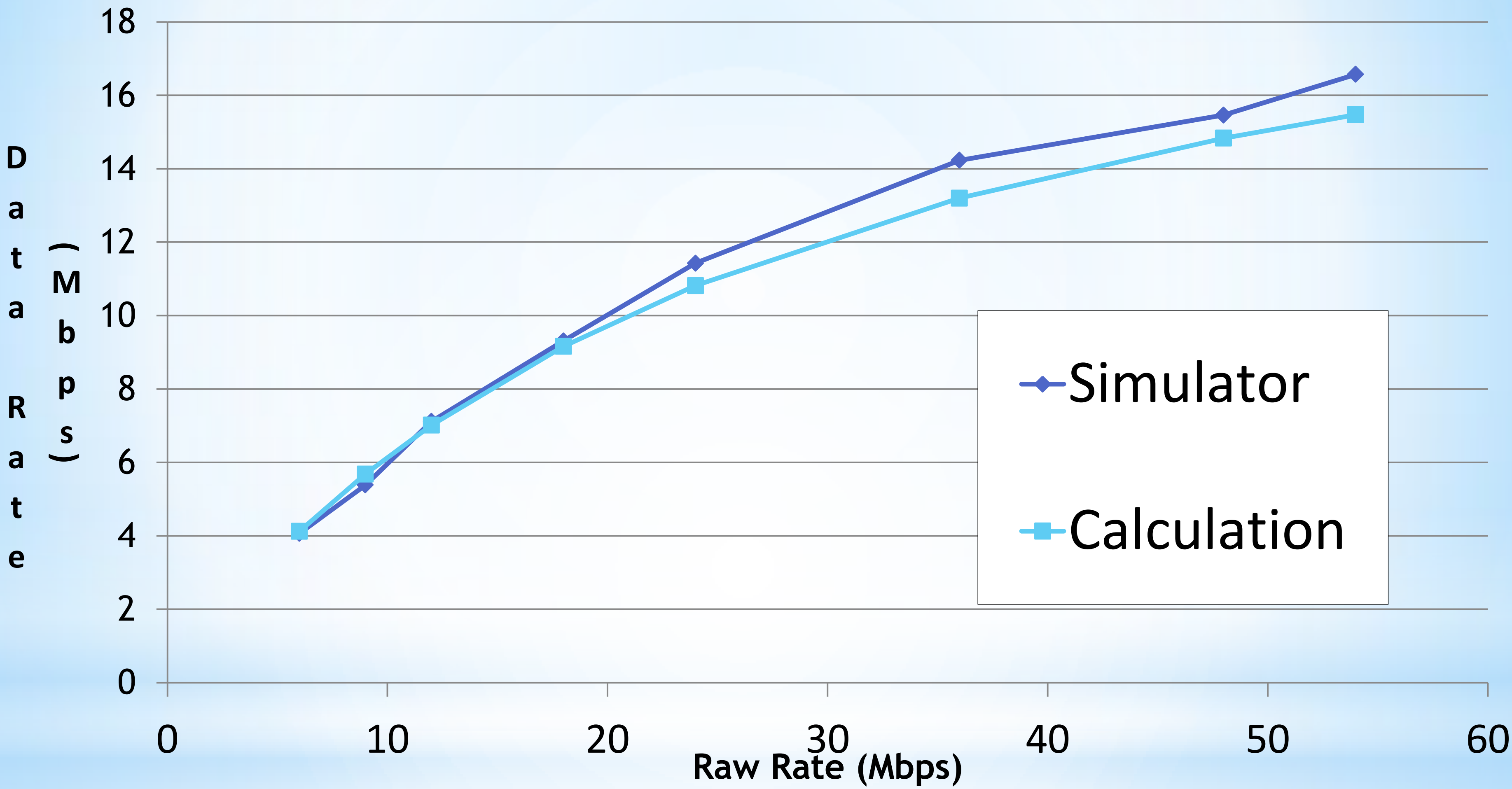
In addition at the end of each such round we drop the oldest packets from $Q(v, s)$, if needed, so that $|Q(v, s)| \leq R_{in}$

Intuitively the algorithms controls the rates and queues using minor changes according to what the wireless channels and queues can sustain.

Packet-Error-Rate / Signal-to-Noise Ratio

- 

* Raw Rate vs. Actual Rate



* Payload size 1KB

* Previous work

- LP(graph) + greedy coloring – constant approximation
- PTAS – impractical due to rounding problem
- Routing in SINR – logarithmic approximation with

uniform power

- simulations in graph model

* Key Contributions

- Adjusted graph model
- SINR simulations
- Path peeling scheduler
- Flow control
- Analysis for Video