SWINBURNE UNIVERSITY OF TECHNOLOGY

Reliable Transport in Delay Tolerant Networks

Manoj Panda

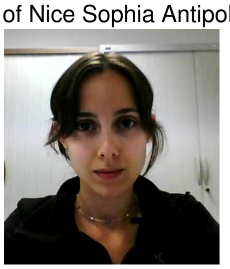
mpanda@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Collaborators/Coauthors

Arshad Ali
(Orange Labs)

Lucile Sassatelli
(Univ. of Nice Sophia Antipolis)



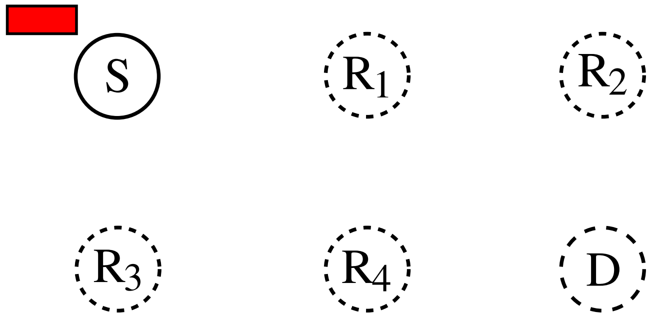
Tijani Chahed
(Telecom SudParis)

Eitan Altman
(INRIA Sophia Antipolis)

[1]
A. Ali, E. Altman, T. Chahed, M. Panda, L. Sassatelli, “A new reliable transport scheme in Delay Tolerant Networks based on acknowledgments and random linear coding,” *23rd International Teletraffic Congress (ITC)*, pp. 214–221, 2011.

Delay Tolerant Networks (DTNs)

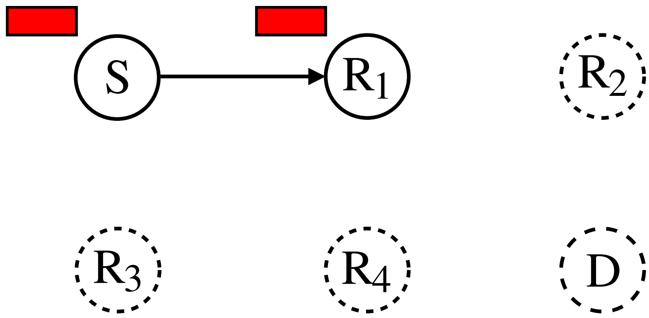
- Sparse network with mostly isolated nodes
- Mostly disconnected - No contemporaneous end-to-end path exists at any time
- Nodes communicate when they come within the radio range
- Requires store-carry-and-forward method of routing



- There can be multiple paths
- Multiple copies might be created inside the network

Delay Tolerant Networks (DTNs)

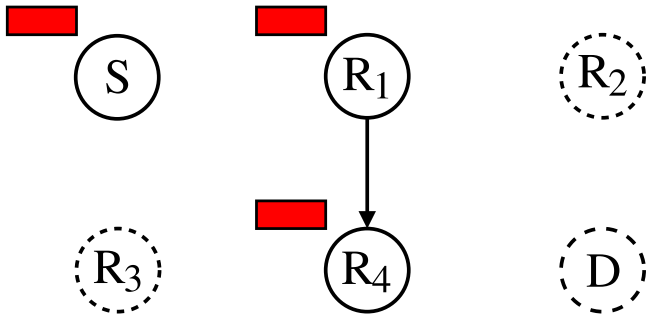
- Sparse network with mostly isolated nodes
- Mostly disconnected - No contemporaneous end-to-end path exists at any time
- Nodes communicate when they come within the radio range
- Requires store-carry-and-forward method of routing



- There can be multiple paths
- Multiple copies might be created inside the network

Delay Tolerant Networks (DTNs)

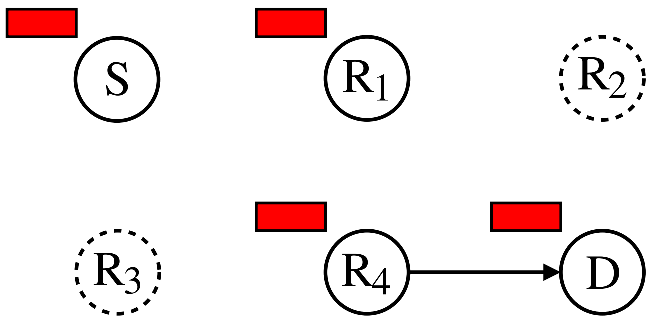
- Sparse network with mostly isolated nodes
- Mostly disconnected - No contemporaneous end-to-end path exists at any time
- Nodes communicate when they come within the radio range
- Requires store-carry-and-forward method of routing



- There can be multiple paths
- Multiple copies might be created inside the network

Delay Tolerant Networks (DTNs)

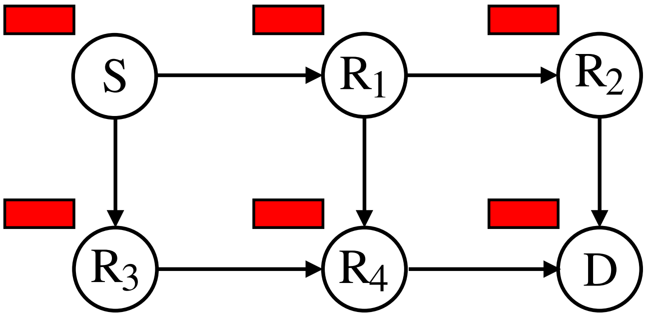
- Sparse network with mostly isolated nodes
- Mostly disconnected - No contemporaneous end-to-end path exists at any time
- Nodes communicate when they come within the radio range
- Requires store-carry-and-forward method of routing



- There can be multiple paths
- Multiple copies might be created inside the network

Delay Tolerant Networks (DTNs)

- Sparse network with mostly isolated nodes
- Mostly disconnected - No contemporaneous end-to-end path exists at any time
- Nodes communicate when they come within the radio range
- Requires store-carry-and-forward method of routing



- There can be multiple paths
- Multiple copies might be created inside the network

Examples of DTNs

- Inter-planetary networks (IPNs)
- Sparse mobile ad hoc networks (MANETs)
- Sparse vehicular ad hoc networks (VANETs)
- Pocket switched networks (MSNs)
- Underwater networks (UWNs)
- Military battlefield networks
- Sensor networks for wildlife tracking
- Remote rural area (village) networks

Our Objective

Design a reliable transport mechanism for DTNs to minimize the end-to-end delivery delay

Literature on Reliable Transport in DTNs

- **Deep-Space**
  - TP-Planet, Bundle Protocol, SCPS-TP, CFDP, LTP, DS-TP, Saratoga, DTTP, ...
- **Terrestrial**
  - PCMP, HIP+TCP timeout and retransmission, ...
- The general approach is to modify TCP
  - Hop-by-hop reliability
  - Selective Negative ACKnowledgments (SNACKs)
  - Host identity and/or connectivity management

Literature on Reliable Transport in DTNs

- **Deep-Space**
  - TP-Planet, Bundle Protocol, SCPS-TP, CFDP, LTP, DS-TP, Saratoga, DTTP, ...
- **Terrestrial**
  - PCMP, HIP+TCP timeout and retransmission, ...
- The general approach is to modify TCP
  - Hop-by-hop reliability
  - Selective Negative ACKnowledgments (SNACKs)
  - Host identity and/or connectivity management

Literature on Reliable Transport in DTNs

- **Deep-Space**
  - TP-Planet, Bundle Protocol, SCPS-TP, CFDP, LTP, DS-TP, Saratoga, DTTP, ...
- **Terrestrial**
  - PCMP, HIP+TCP timeout and retransmission, ...
- The general approach is to modify TCP
  - Hop-by-hop reliability
  - Selective Negative ACKnowledgments (SNACKs)
  - Host identity and/or connectivity management

SWINBURNE UNIVERSITY OF TECHNOLOGY Caia Seminar http://www.caia.swin.edu.au mpanda@swin.edu.au July 4, 2013 6

Our Approach

- We propose a framework rather than a detailed protocol

Our Approach

- We propose a framework rather than a detailed protocol
- The framework
  - Propose a reliable transport mechanism
  - Develop a fluid-limit analytical model
  - Optimize the parameters based on the analytical model

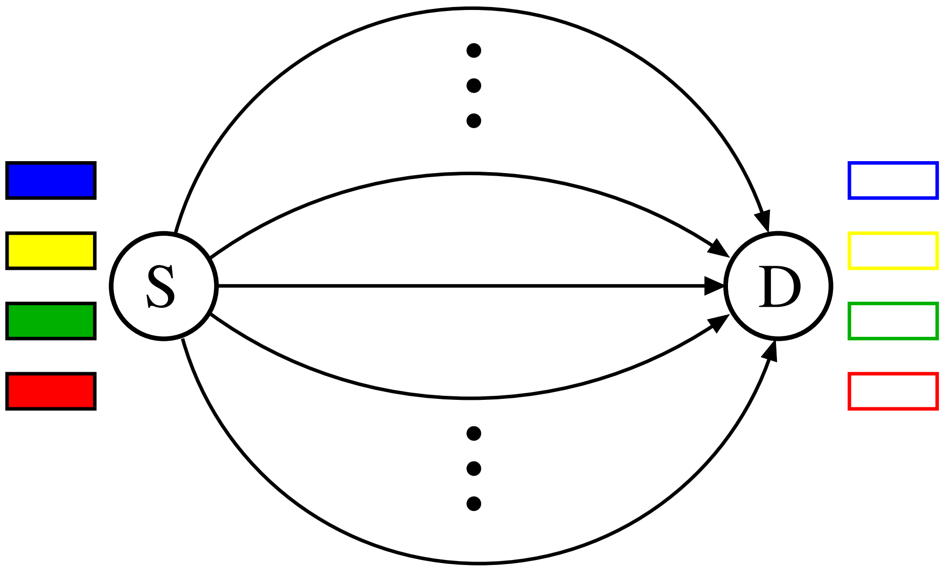
Our Approach

- We propose a framework rather than a detailed protocol
- The framework
  - Propose a reliable transport mechanism
  - Develop a fluid-limit analytical model
  - Optimize the parameters based on the analytical model
- Our transport mechanism is similar to Sundararajan et al [2]
  - Random Linear Coding at the source
  - Degree of Freedom (DoF) Acknowledgments (ACKs)

[2] J.K. Sundararajan, D. Shah, M. Medard, S. Jacubczak, M. Mitzenmacher, J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proceedings of the IEEE*, Vol. 99, No. 3, March 2011.

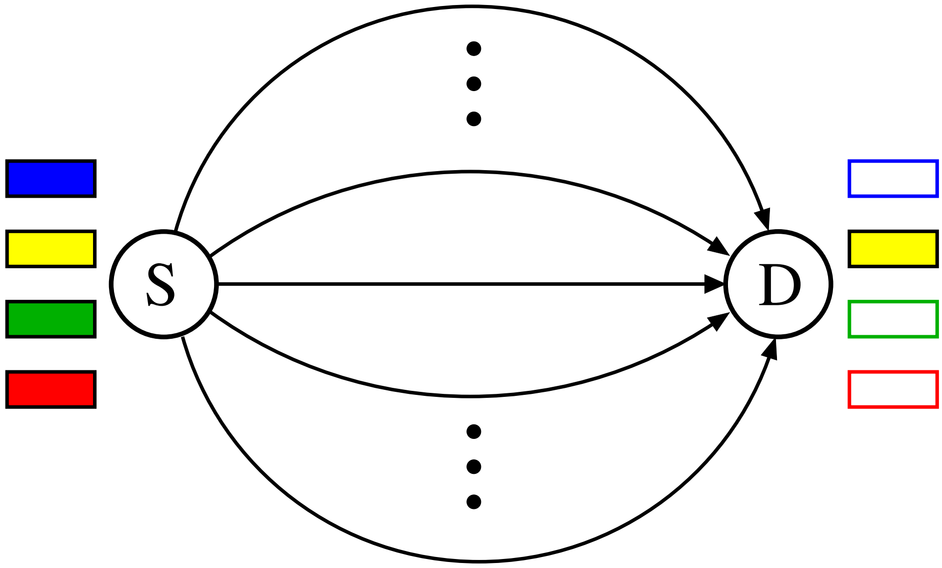
Transport With and Without Coding

TRANSPORT WITHOUT CODING



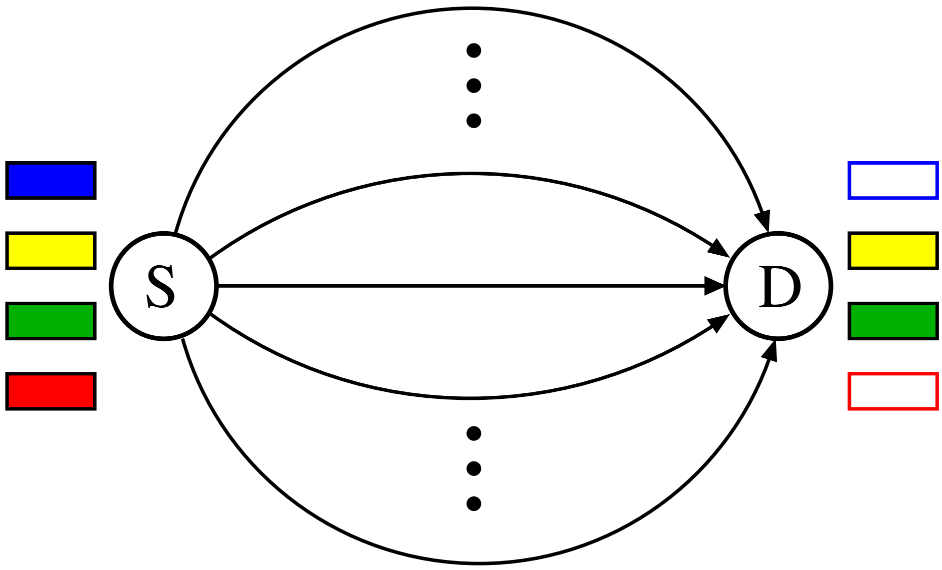
Transport With and Without Coding

TRANSPORT WITHOUT CODING

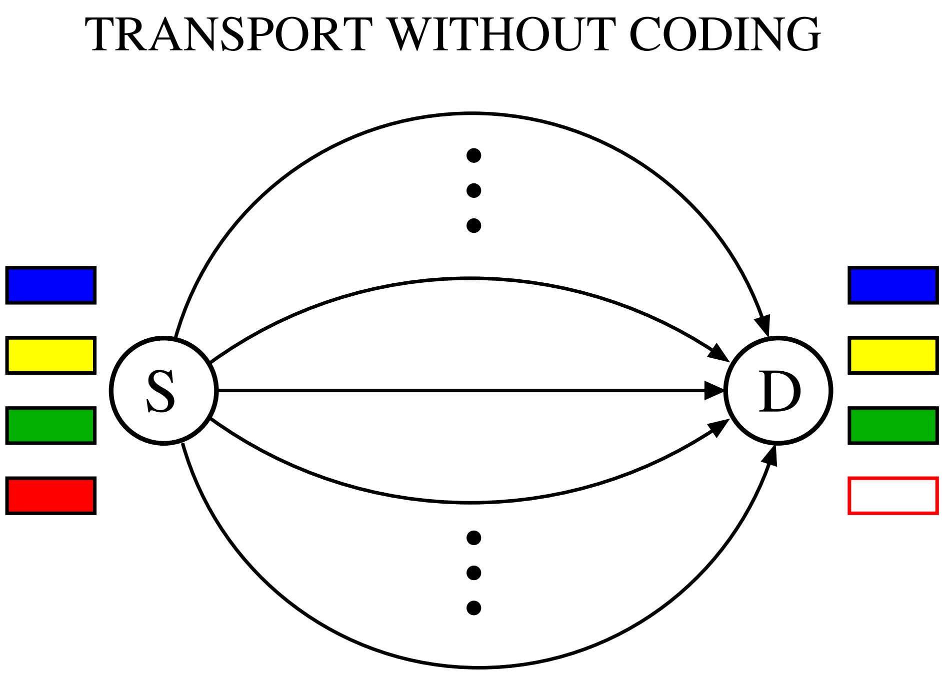


Transport With and Without Coding

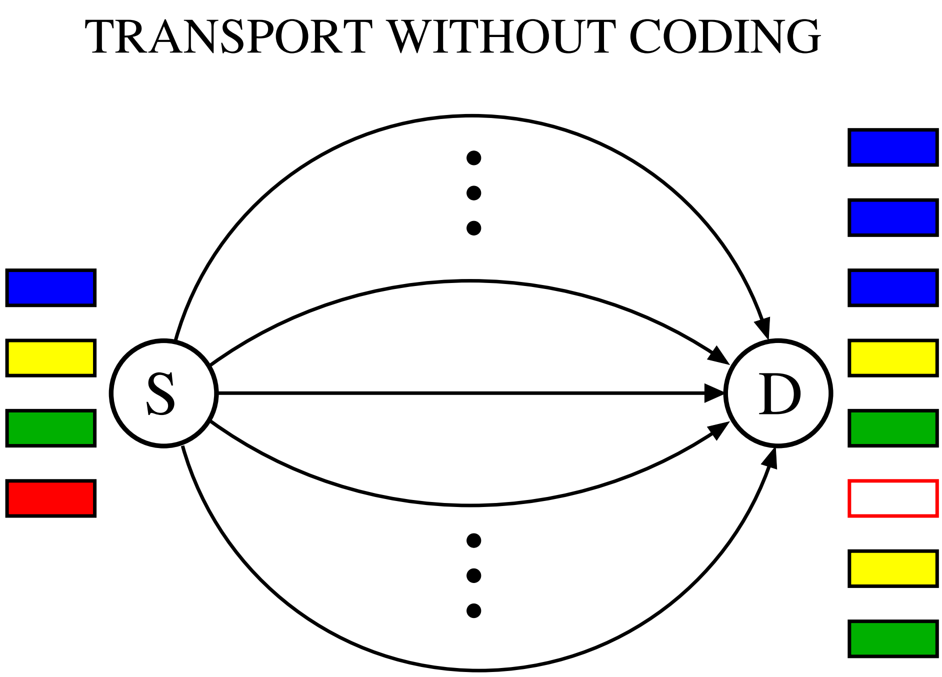
TRANSPORT WITHOUT CODING



Transport With and Without Coding



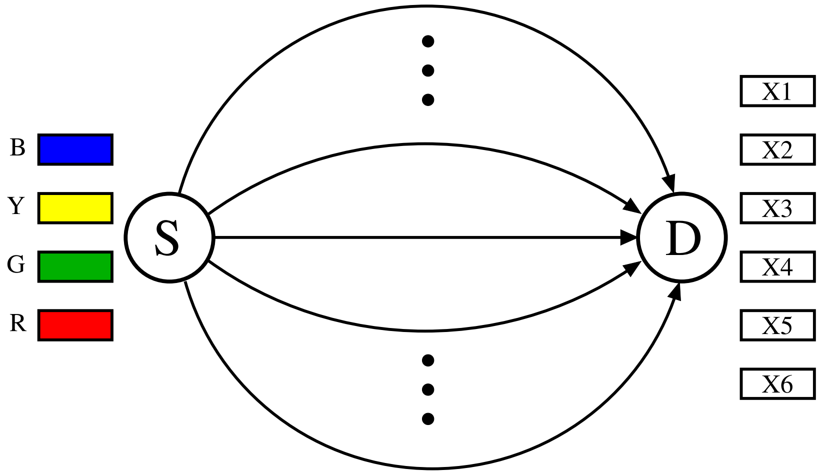
Transport With and Without Coding



Caia Seminar http://www.caia.swin.edu.au mpanda@swin.edu.au July 4, 2013 8

Transport With and Without Coding

TRANSPORT WITH CODING



X1 = a1 B + b1 Y + c1 G + d1 R X2 = a2 B + b2 Y + c2 G + d2 R
X3 = a3 B + b3 Y + c3 G + d3 R X4 = a4 B + b4 Y + c4 G + d4 R
X5 = a5 B + b5 Y + c5 G + d5 R X6 = a6 B + b6 Y + c6 G + d6 R

Degrees of Freedom (DoF)

Destination receives $\mathbf{x} = [X1 \quad X2 \quad X3 \quad X4 \quad \ldots]^T$

Coding Vector for $X1$ is $[a1 \quad b1 \quad c1 \quad d1]$

$$\text{Decoding Matrix } \mathbf{D} = \begin{bmatrix} a1 & b1 & c1 & d1 \\ a2 & b2 & c2 & d2 \\ a3 & b3 & c3 & d3 \\ a4 & b4 & c4 & d4 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

D and **x** grow with time

Coded packets satisfy $\boxed{\mathbf{x} = \mathbf{D}\mathbf{y}}$ where $\mathbf{y} = [B \quad Y \quad G \quad R]^T$

Unique solution to **y** when **D** attains full rank

The source needs to receive as many Degrees of Freedom (DoF) as the number of original packets

Missing DoF = Number of native/original packets − Rank of **D**

ACKs carry the missing DoF at the destination

Outline

Our Proposed Transport Framework

Analytical Modeling

Optimization

Outline

Our Proposed Transport Framework

Analytical Modeling
- Model Validation

Optimization
- Optimal Mean Delivery Delay

Network Setting and Assumptions

- **Network Setting**
  - We have one source-destination pair and a few relays
  - $M =$ Number of native/original packets to be sent

Network Setting and Assumptions

- **Network Setting**
  - We have one source-destination pair and a few relays
  - $M =$ Number of native/original packets to be sent
- **Assumptions**
  - A relay stores at most one data packet or ACK
  - Drops packets from buffer after an **expiry timeout**

Network Operation

- **Source-Relay Meetings**
  - The source generates a Random Linear Combination (RLC)
  - The coding vector is added to the header of coded packet
  - The relay gets the coded packet (or RLC)

Network Operation

- **Source-Relay Meetings**
  - The source generates a Random Linear Combination (RLC)
  - The coding vector is added to the header of coded packet
  - The relay gets the coded packet (or RLC)

- **Relay-Relay Meetings:** Epidemic-Type Replication
  - The coded packet is copied into the other relay

Network Operation

- **Source-Relay Meetings**
  - The source generates a Random Linear Combination (RLC)
  - The coding vector is added to the header of coded packet
  - The relay gets the coded packet (or RLC)
- **Relay-Relay Meetings:** Epidemic-Type Replication
  - The coded packet is copied into the other relay
- **Destination-Relay Meetings**
  - The destination gets the coded packet
  - Updates the missing DoF and generates an ACK
  - The relay gets the ACK

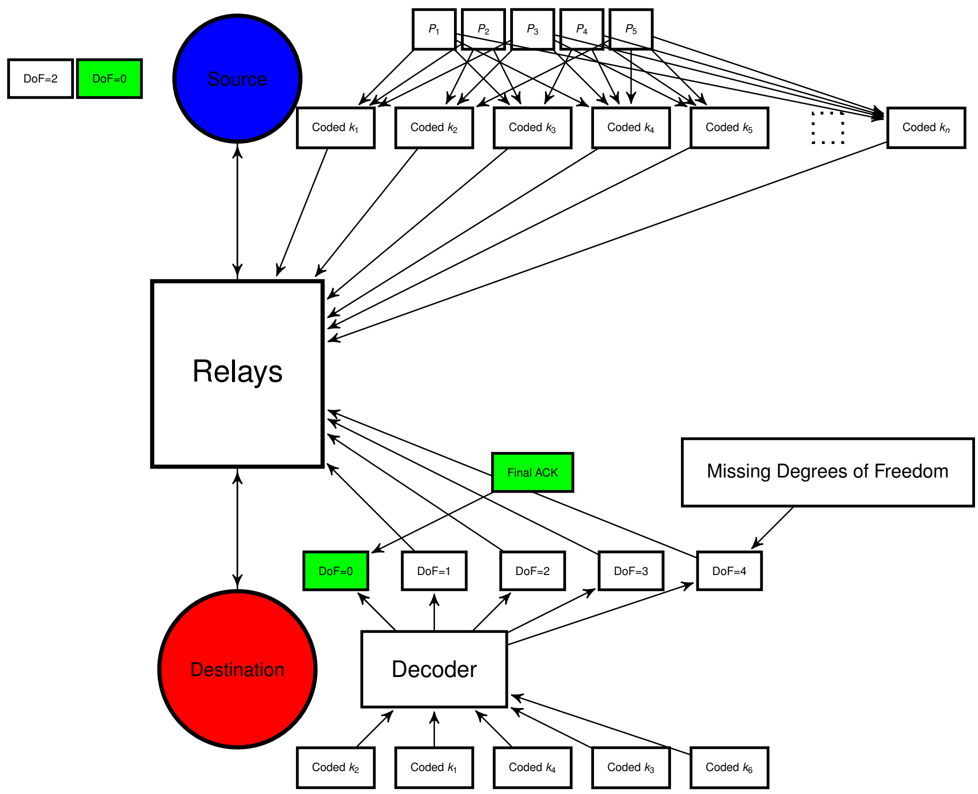
Network Operation

- **Source-Relay Meetings**
  - The source generates a Random Linear Combination (RLC)
  - The coding vector is added to the header of coded packet
  - The relay gets the coded packet (or RLC)
- **Relay-Relay Meetings:** Epidemic-Type Replication
  - The coded packet is copied into the other relay
  - The ACK is copied into the other relay
- **Destination-Relay Meetings**
  - The destination gets the coded packet
  - Updates the missing DoF and generates an ACK
  - The relay gets the ACK

Network Operation

- **Source-Relay Meetings**
  - The source generates a Random Linear Combination (RLC)
  - The coding vector is added to the header of coded packet
  - The relay gets the coded packet (or RLC)
  - The source gets the ACK and updates the missing DoF
- **Relay-Relay Meetings:** Epidemic-Type Replication
  - The coded packet is copied into the other relay
  - The ACK is copied into the other relay
- **Destination-Relay Meetings**
  - The destination gets the coded packet
  - Updates the missing DoF and generates an ACK
  - The relay gets the ACK

Network Operation with Coded Packets



What can go wrong?

- **The ACKs may not find enough relay buffer**

What can go wrong?

- **The ACKs may not find enough relay buffer**
  - Buffer expiry might help

What can go wrong?

- **The ACKs may not find enough relay buffer**
  - Buffer expiry might help
  - Limit the **Number of Coded Packets**
  - Limit the duration of replication - **Spreading Time**
  - Wait for the ACKs

What can go wrong?

- **The ACKs may not find enough relay buffer**
  - Buffer expiry might help
  - Limit the **Number of Coded Packets**
  - Limit the duration of replication - **Spreading Time**
  - Wait for the ACKs
- **There can be deadlock**

What can go wrong?

- **The ACKs may not find enough relay buffer**
  - Buffer expiry might help
  - Limit the **Number of Coded Packets**
  - Limit the duration of replication - **Spreading Time**
  - Wait for the ACKs
- **There can be deadlock**
  - Limited coded packets
  - First few coded packets replicate much faster
  - Coded packets generated later do not find enough relay buffer or are dropped faster

What can go wrong?

- **The ACKs may not find enough relay buffer**
  - Buffer expiry might help
  - Limit the **Number of Coded Packets**
  - Limit the duration of replication - **Spreading Time**
  - Wait for the ACKs
- **There can be deadlock**
  - Limited coded packets
  - First few coded packets replicate much faster
  - Coded packets generated later do not find enough relay buffer or are dropped faster

$\Rightarrow$ Relay buffers finally filled with ACKs with missing DoF > 0

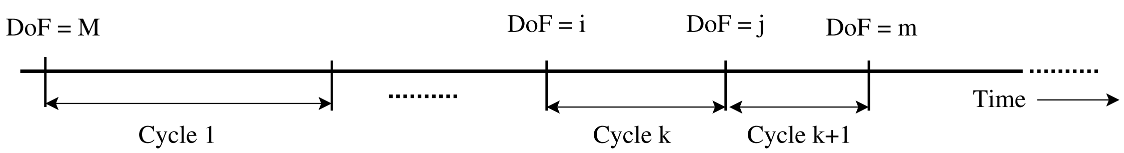
What can go wrong?

- **The ACKs may not find enough relay buffer**
  - Buffer expiry might help
  - Limit the **Number of Coded Packets**
  - Limit the duration of replication - **Spreading Time**
  - Wait for the ACKs
- **There can be deadlock**
  - Limited coded packets
  - First few coded packets replicate much faster
  - Coded packets generated later do not find enough relay buffer or are dropped faster

    $\Rightarrow$ Relay buffers finally filled with ACKs with missing DoF > 0

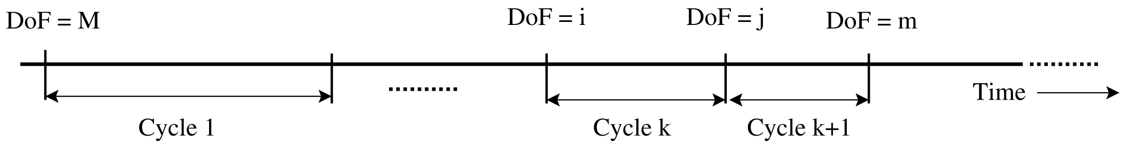
  - Limit the duration of waiting for ACKs - **ACK Waiting Time**
  - Drop everything after some time and restart - **Cycle Time**

Our Proposal Operates in Cycles



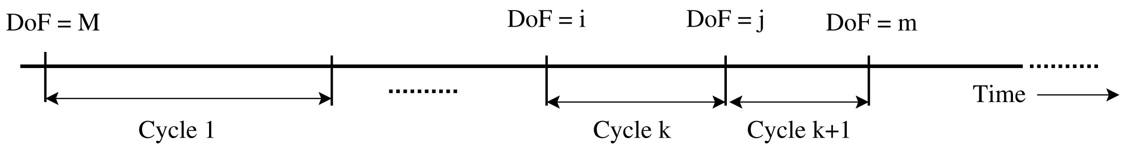
- Drop everything from time to time and restart

Our Proposal Operates in Cycles



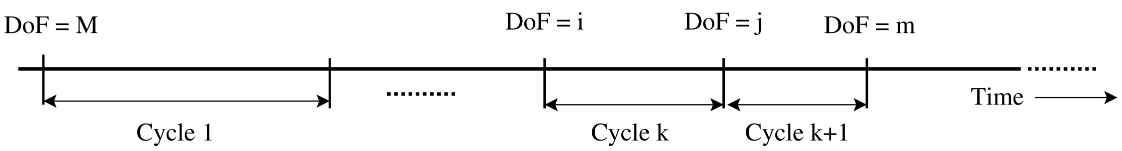
- Drop everything from time to time and restart
- These are checkpoints to monitor the progress
  - Missing DoF **i** from the point of view of the source

Our Proposal Operates in Cycles



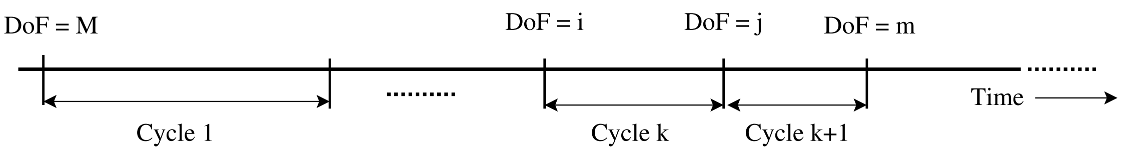
- Drop everything from time to time and restart
- These are checkpoints to monitor the progress
  - Missing DoF **i** from the point of view of the source
- Time between two checkpoints is called a cycle

Our Proposal Operates in Cycles



- Drop everything from time to time and restart
- These are checkpoints to monitor the progress
  - Missing DoF **i** from the point of view of the source
- Time between two checkpoints is called a cycle
- Based on the missing DoF **i** at the beginning of the cycle **choose**
  - Number of coded packets, M_i
  - Spreading time, $\tau_{i,S}$
  - Waiting time, $\tau_{i,W}$

Our Proposal Operates in Cycles



- Drop everything from time to time and restart
- These are checkpoints to monitor the progress
  - Missing DoF **i** from the point of view of the source
- Time between two checkpoints is called a cycle
- Based on the missing DoF **i** at the beginning of the cycle **choose**
  - Number of coded packets, M_i
  - Spreading time, $\tau_{i,S}$
  - Waiting time, $\tau_{i,W}$
- Continue until the source receives the final ACK

Structure of a Cycle

Suppose that the cycle begins with missing DoF i

- Control Parameters
  - M_i = Number of coded packets to be sent
  - $\tau_{i,S}$ = Spreading time
  - $\tau_{i,W}$ = ACK wait time
- t_k = Time of generation of coded packet k, $1 \leq k \leq M_i$
- Each coded packet is replicated for a duration $\tau_{i,S}$
- E.g., the coded packet k is replicated during $(t_k, t_k + \tau_{i,S}]$
- Cycle duration:

$$\tau_i := t_{M_i} + \tau_{i,S} + \tau_{i,W}$$

- There can be up to $i - 1$ ACKs indicating missing DoF = $i - 1, \ldots, 0$

Outline

Our Proposed Transport Framework

Analytical Modeling

Model Validation

Optimization

Optimal Mean Delivery Delay

Network Parameters

- 1 sources, 1 destination, N_0 relays

Network Parameters

- 1 sources, 1 destination, N_0 relays
- Exponentially distributed pairwise inter-meeting times

Network Parameters

- 1 sources, 1 destination, N_0 relays
- Exponentially distributed pairwise inter-meeting times
  - $\beta_s =$ Rate of meeting between source and a relay
  - $\beta_d =$ Rate of meeting between destination and a relay
  - $\beta_r =$ Rate of meeting between two relays

Network Parameters

- 1 sources, 1 destination, N_0 relays
- Exponentially distributed pairwise inter-meeting times
  - $\beta_s =$ Rate of meeting between source and a relay
  - $\beta_d =$ Rate of meeting between destination and a relay
  - $\beta_r =$ Rate of meeting between two relays
- Exponentially distributed buffer expiry time
  - $\tau_e = 1/\beta_e =$ Mean time to buffer expiry

Fluid-Limit Equations

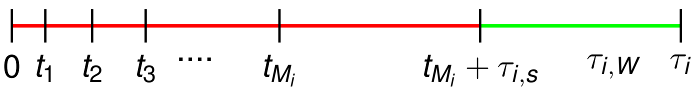
Each cycle can be modeled independently

Fluid-Limit Equations

Each cycle can be modeled independently

Coded packets replication

- $x_k(t)$ = fraction of relay nodes having the coded packet k
- $y_l(t)$ = fraction of relay nodes having the ACK l
- $x(t) := \sum_k x_k(t)$, $y(t) := \sum_k y_k(t)$
- $q_k(t)$ = Pr($k - 1$ source-relay meetings up to t)
- $\lambda_s = N_0\beta_s$, $\lambda_d = N_0\beta_d$, $\lambda_r = N_0\beta_r$, $s = d = 1/N_0$



$$\frac{dx_k(t)}{dt} = \begin{cases} (s\lambda_s q_k(t) + \lambda_r x_k(t))(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t), \quad \text{for} \quad 0 \le t \le \frac{k}{\lambda_s} + \tau_{i,S}, \\ \\ -d\lambda_d x_k(t) - \beta_e x_k(t), \quad \text{for} \quad \frac{k}{\lambda_s} + \tau_{i,S} < t \le \tau_i. \end{cases}$$

Initial Condition: $x_k(0) = 0$, for all $1 \le k \le M_i$

Fluid-Limit Equations

ACKs replication

- $P_{X_k}(t)$ = Probability that the destination has received coded packet k by time t
- $Q_l^{(i)}(t)$ = Prob. that exactly $i-l$ DoFs have been received by the destination in the current cycle by time t

$$Q_l^{(i)}(t) = \sum_{E\subset\{1,\ldots,M_i\}\,:\,|E|=i-l} \prod_{m\in E} P_{X_m}(t) \prod_{m'\in\{1,\ldots,M_i\}\setminus E} (1-P_{X_{m'}}(t)).$$

$$\frac{dy_l(t)}{dt} = \lambda_r y_l(t)(1-x(t)-y(t)) + d\lambda_d Q_l^{(i)}(t)(1-y_l(t)) \quad \text{for} \quad 0 < t \le \tau_i.$$

Initial Condition: $y_l(0) = 0$, for all $0 \le l \le i-1$

Fluid-Limit Equations

Equations for the CDFs of Packet and ACK Delay

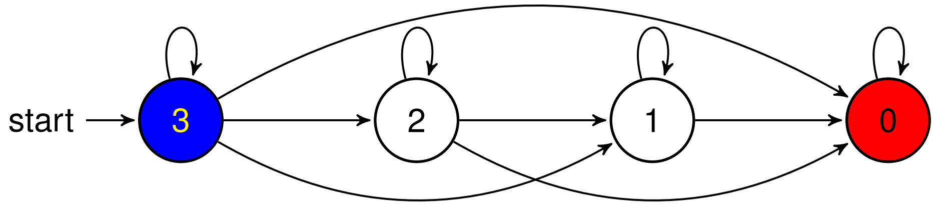
- $P_{X_k}(t)$ = Probability that the destination has received coded packet k by time t
- $P_{Y_l}(t)$ = Probability that the source has received ACK l by time t

$$\frac{dP_{X_k}(t)}{dt} = \lambda_d x_k(t)(1-P_{X_k}(t))$$

$$\frac{dP_{Y_l}(t)}{dt} = \lambda_s y_l(t)(1-P_{Y_l}(t))$$

Initial Condition: $P_{X_k}(0) = 0$, for all $1 \le k \le M_i$,
$P_{Y_l}(0) = 0$, for all $0 \le l \le i-1$.

Combining the Sequence of Cycles



State: Missing DoF at source at the beginning of cycle
P_{ij} = Transition probability from state i to state j, $j = i-1, \dots, 0$

$$P_{ij} \quad = \quad P_{Y_j}(\tau_i) \prod_{l=0}^{j-1} (1 - P_{Y_l}(\tau_i)), \quad \text{for } j < i \text{ and } \quad P_{ii} = 1 - \sum_{j=0}^{i-1} P_{ij}.$$

T_i = mean time to reach state 0 starting from state i

$$T_i = \frac{\sum_{j=1}^{i-1} P_{ij} T_j + \tau_i - \int_0^{\tau_i} P_{i0}(t) dt}{1 - P_{ii}(\tau_i)}.$$

$T_M =$ **Mean Delivery Delay** or **Mean File Transfer Time**

Outline

Our Proposed Transport Framework

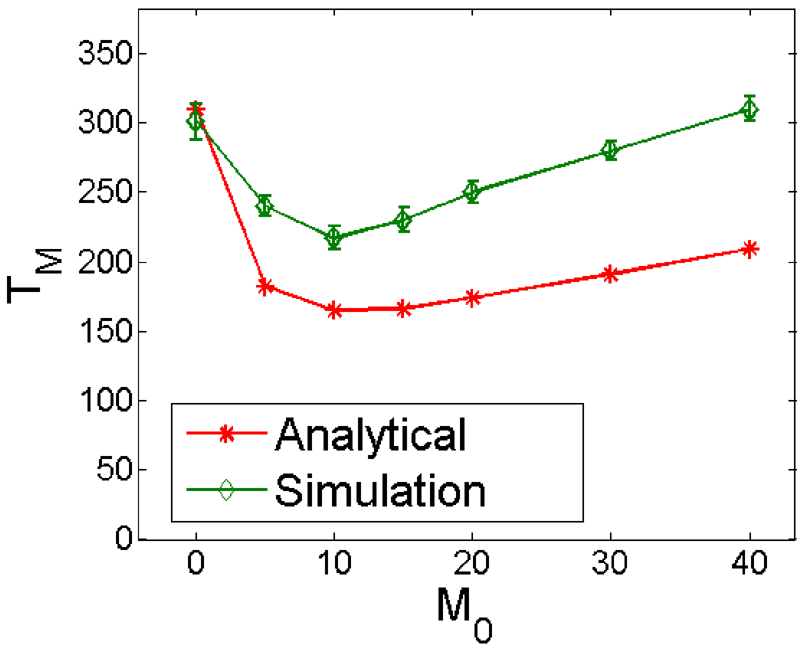
Analytical Modeling
- Model Validation

Optimization
- Optimal Mean Delivery Delay

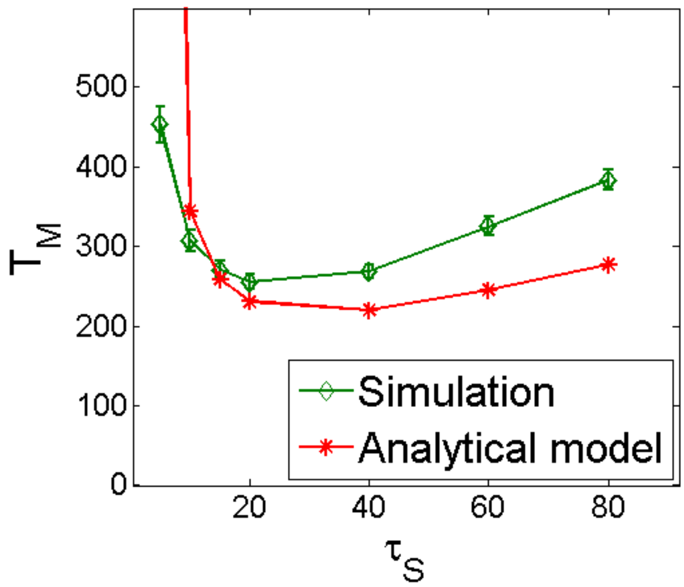
Parameter Setting

- Number of packets to be transferred $M = 10$
- Number of relay nodes $N_0 = 100$
- Inter meeting rate $\beta_s = \beta_d = \beta_r = \beta = 0.005$
- $\tau_{i,S} = \tau_S$, $\tau_{i,W} = \tau_W$, $M_i = M_0 + i$
- M_0 : Level of redundancy
- $\tau_S = \tau_W = \tau_e = 10$, $M_0 = 0, 20$
- Developed a Matlab based simulation
- Random **binary** coefficients are used
- Simulation results averaged over 1000 runs

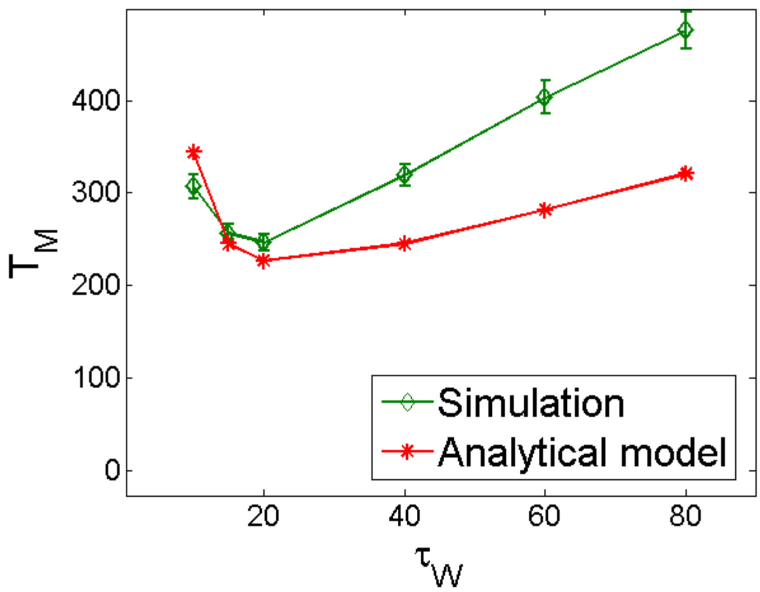
Mean Delivery Delay, T_M, vs. Redundancy, M_0



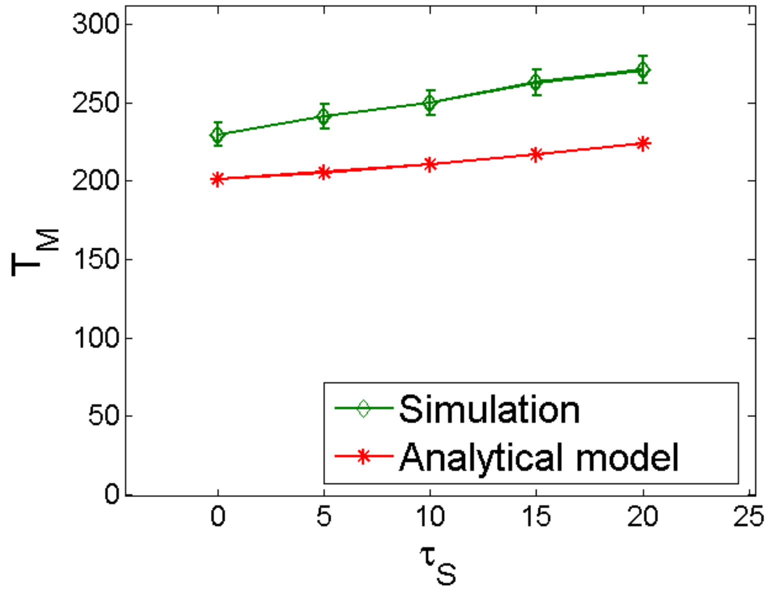
Mean Delivery Delay, T_M, vs. Spread Time, τ_S $M_0 = 0$



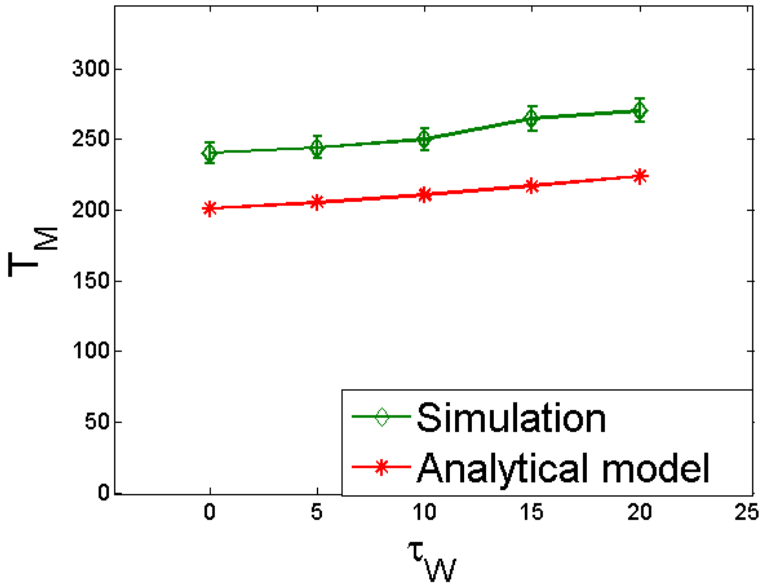
Mean Delivery Delay, T_M, vs. Wait Time, τ_W $M_0 = 0$



Mean Delivery Delay, T_M, vs. Spread Time, τ_S $M_0 = 20$



Mean Delivery Delay, T_M, vs. Wait Time, τ_W $M_0 = 20$



Outline

Our Proposed Transport Framework

Analytical Modeling
- Model Validation

Optimization
- Optimal Mean Delivery Delay

Optimization Procedure

Objective

Minimize the mean time to transfer the complete file, i.e., T_M

Optimization over parameters
- M_i = Number of coded packets to be sent
- $\tau_{i,S}$ = Spreading time
- $\tau_{i,W}$ = ACK wait time

$\forall i,\ i = 1, 2, \ldots, M,\ \forall j,\ i = 0, 1, \ldots, i - 1$

$$\min(T_i) = \frac{\sum_{j=1}^{i-1} P_{ij} \min(T_j) + \tau_i - \int_0^{\tau_i} P_{i0}(t)dt}{1 - P_{ii}(\tau_i)}.$$

Outline

Our Proposed Transport Framework

Analytical Modeling
- Model Validation

Optimization
- Optimal Mean Delivery Delay

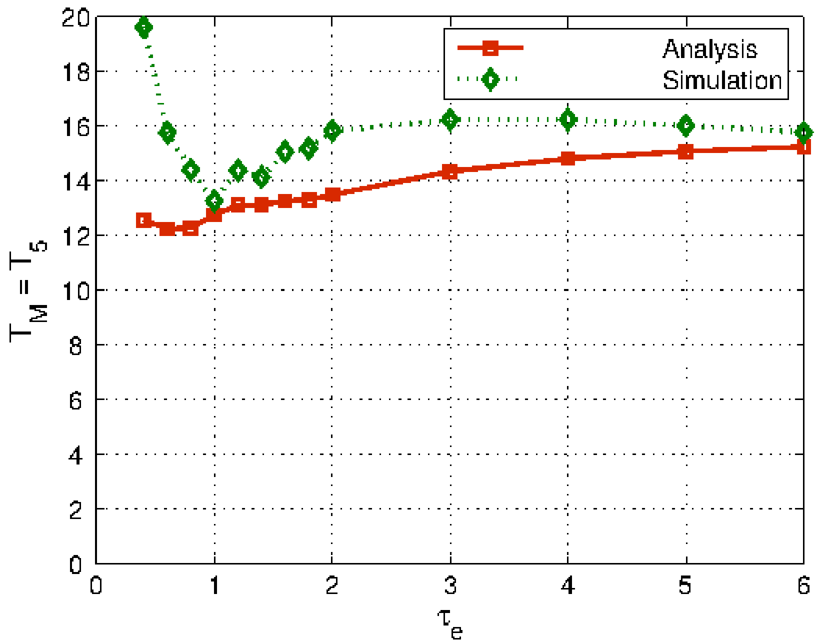
Parameter Setting

- Number of packets to be transferred $M = 5$
- Number of relay nodes $N_0 = 100$
- Inter meeting rate $\beta_s = \beta_d = \beta_r = \beta = 0.05$
- Buffer expiry timeout is varied

We apply **Differential Evolution (DE)** [3] for optimization

[3] R. Storn, K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol. 11, pp. 341–359,1997.

Optimal Mean Delivery Delay Vs. Expiry Time



Conclusion

- We proposed a reliable transport framework
  - Random Linear Coding
  - Degree of Freedom (DoF) ACKs
- Derived a fluid-limit model for a single flow
- Optimized the parameters using DE
- Future Work
  - Multiple interacting flows
  - Multiple buffer and buffer management
  - Non-exponential inter-meeting times

Thank You

Timer Implementation

- Let M, N_0, β_s, β_r, β_d be known
- Compute M_i, $\tau_{i,S}$ and $\tau_{i,W}$ off-line for each i
- At source-relay meeting t_k, compute
  - Spreading timeout $t_k + \tau_{i,S}$
  - Cycle timeout $\tau_i = (M_i/\lambda_s) + \tau_{i,S} + \tau_{i,W}$
  - Store in the header of the coded packet
  - Buffer expiry timeout $t_k + \exp(\beta_e)$
- The relay drops the coded packet at $t_k + \exp(\beta_e)$ or τ_i
- At relay-relay meeting t'_k, copy the coded packet to other relay if $t'_k < t_k + \tau_{i,S}$
- At destination-relay meeting, copy the cycle timeout τ_i to the header of the ACK