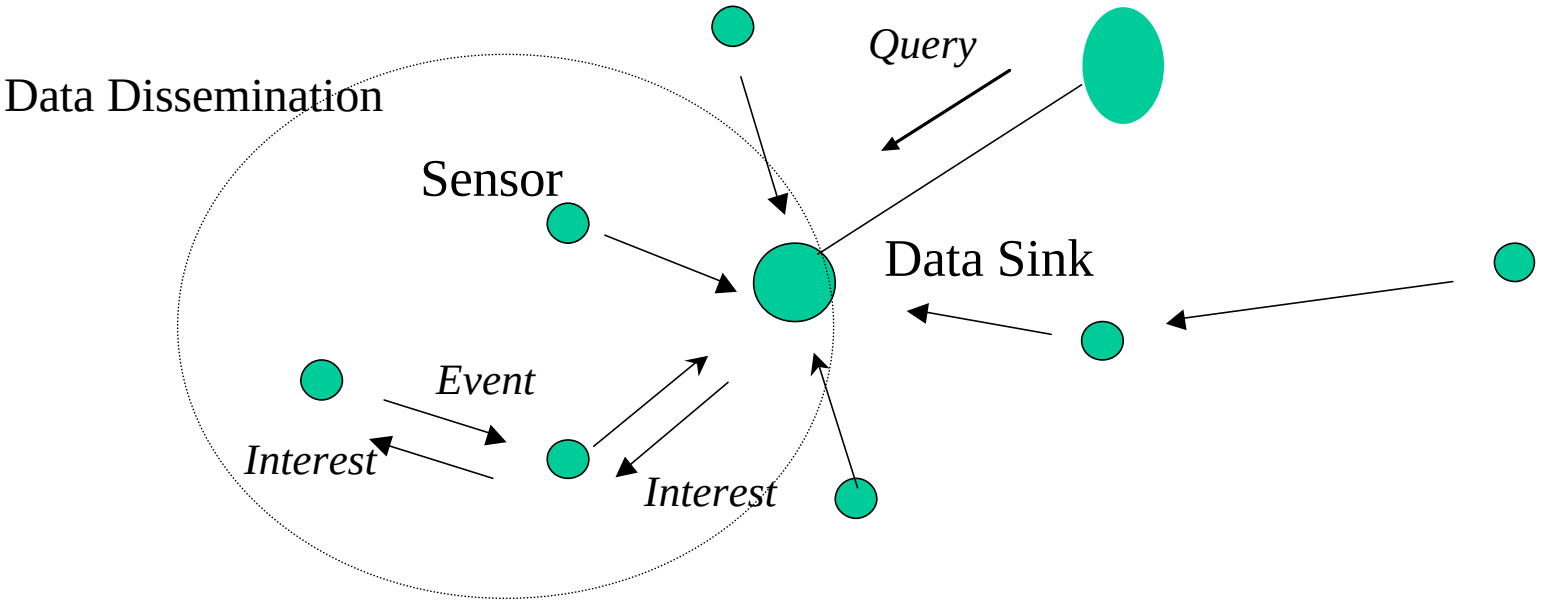
# PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks

**Chieh-Yih Wan, Andrew T. Campbell**
**Comet Group, Columbia University**

**Lakshman Krishnamurthy**
**Intel Research and Development**
**Intel Corporation**

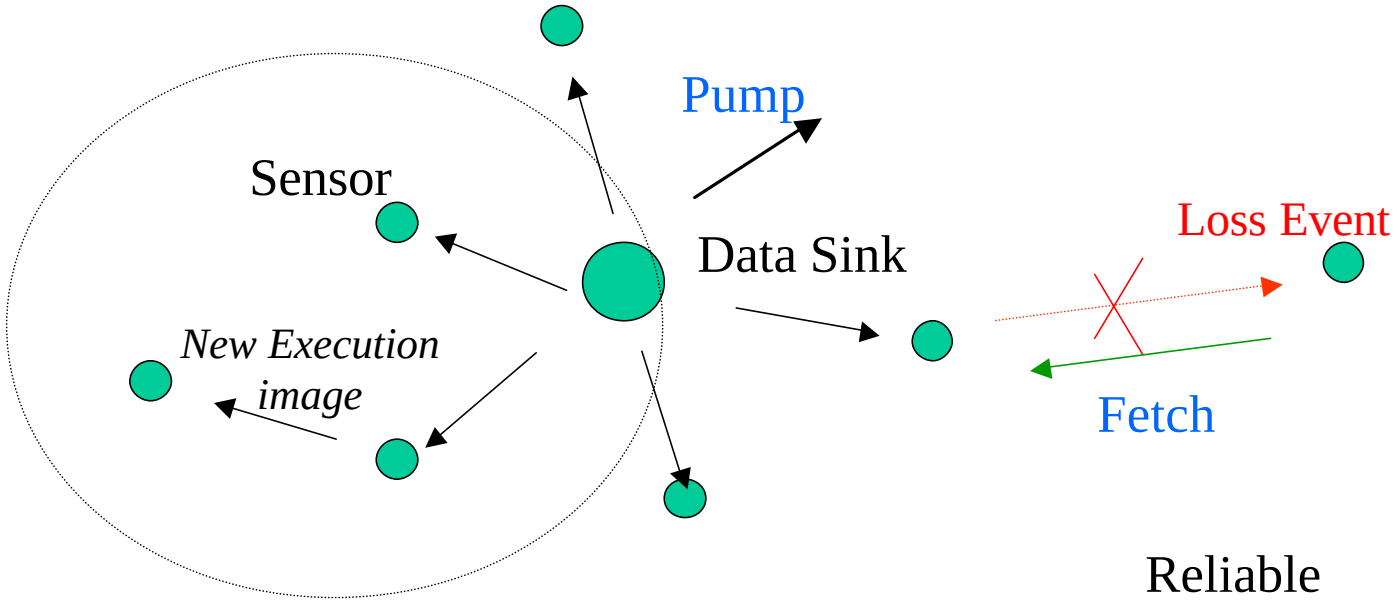
**First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA' 2002), Atlanta, September 28, 2002**

# Traditional Viewpoint: "Sources-to-Sink Communications"




Occasional loss is OK

# New Viewpoint: "Sink-to-Sources" Communications




Reliable Multicast?

Occasional loss is disastrous

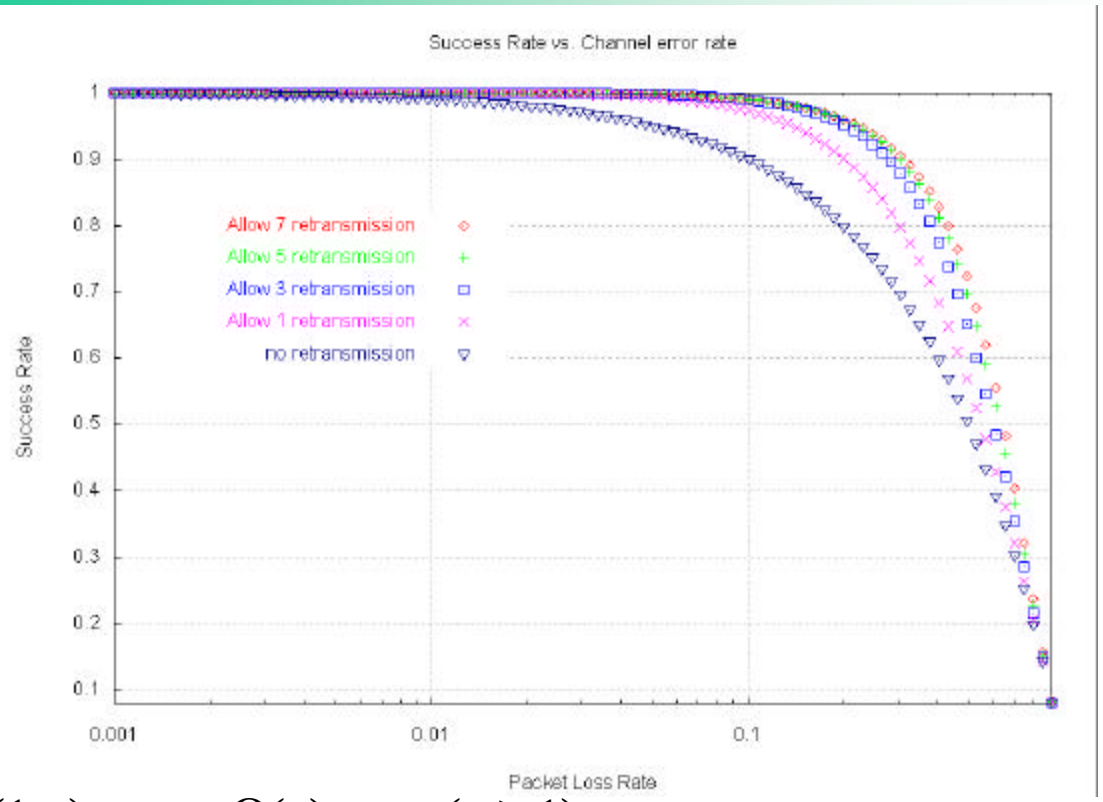
# “Pump Slowly, Fetch Quickly”

- Ensure delivery with min. support from infrastructure ~ non-IP.
- Minimal signaling – lost detection, recovery.
- High Error tolerance.

# PSFQ Features

- Negative ACK system
- Hop by hop error recovery – not e2e
- 1->N reliable delivery
- Fundamental relationship between pump and fetch
- Multi-modal communications operation

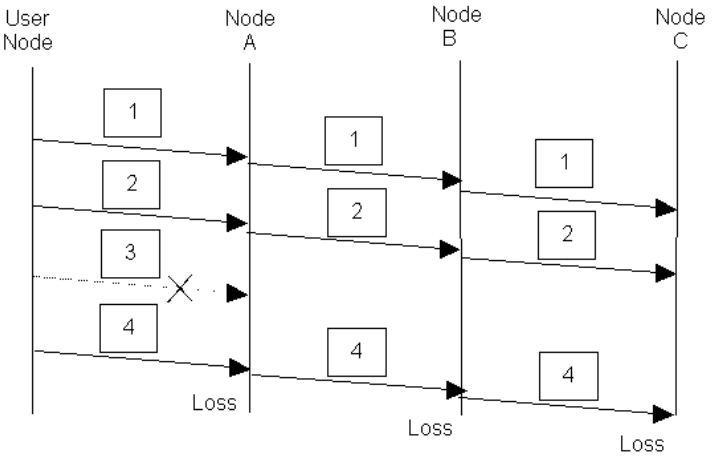
# Fetch/Pump Relationship

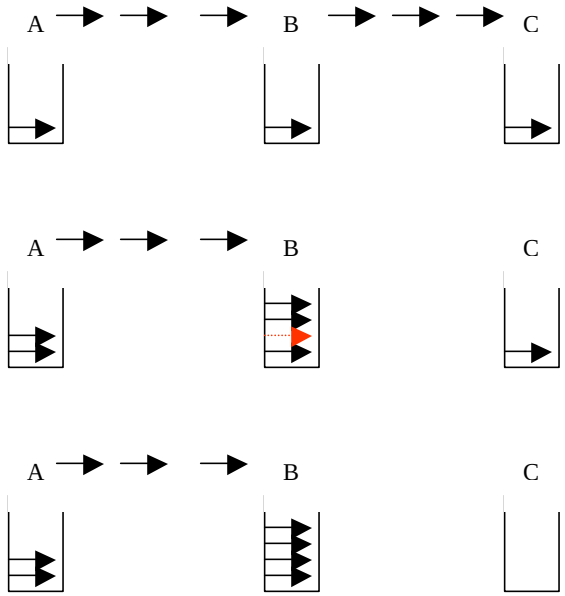



- $(1-p) + p \times \Omega(n) \qquad (n \geq 1)$
- $\Omega(n) = \Phi(1) + \Phi(2) + \ldots + \Phi(n)$
- $\Phi(n) = (1-p)^2 \times [1 - p - \Phi(1) - \Phi(2) - \ldots - \Phi(n-1)] \quad \Phi(0) = 0$

# Multi-Modal Operations

- “Multihop forwarding” vs. “Store-and-forward”.
- Propagation of Loss Event




- In-sequence data forwarding.

# PSFQ Operations

- Pump
  - Timers – $T_{min}$, $T_{max}$
- Fetch
  - Timers
    - Tr (<< Tmax)
    - Tmax/Tr ~ Fetch / Pump ratio.
  - Loss Aggregation – windows of loss.
- Report

# Pump Operation

- Timers – $T_{min}$, $T_{max}$
- $T_{min}$
  - Time-buffer for local recovery
  - Suppress rebroadcast [mobicom99]
- $T_{max}$ – loose delay bounds
  $D(n) = T_{max} \times n \times (Number\ of\ hops)$

# Fetch Operation

- Loss Aggregation – windows of loss.
- Timers
  - Tr (<< Tmax)
  - Tmax/Tr ~ Pump/Fetch ratio.
- Proactive Fetch
  - Loss of last segment.
  - Loss of all segment.
  - How long should wait before proactive fetch?

# Fetch Operation

- Proactive Fetch
  - Correct choice of $T_{pro}$
  - $T_{pro} = a * (S_{max} - S_{last}) * T_{max}$ ($a \geq 1$). (no limitation on cache size)
  - $T_{pro} = a * n * T_{max}$ ($a \geq 1$). (data cache keeps only n segments)

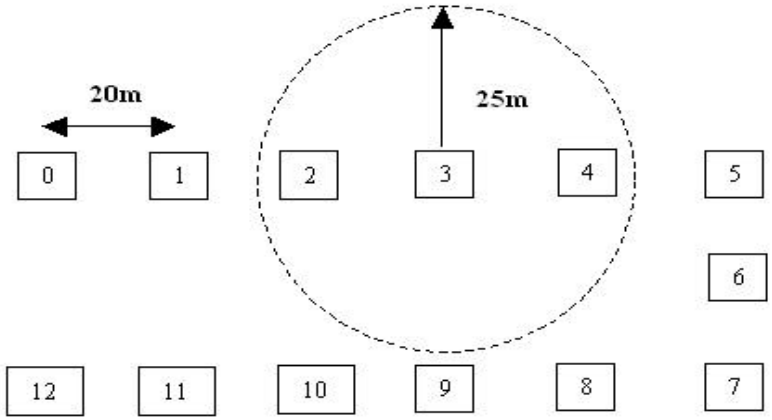
# Report Operation

- Soliciting report.
- Piggybacking report operation on hop-by-hop basis.
- Report timers
  - $T_{report} = T_{max} \times TTL + ?$ .

# Performance Evaluation

- Compare with SRM.
  - Three control mesg.: session, request and repair.
- Idealized SRM – extract out IP multicast substrate, replace with Omniscient multicast.
- Performance Metrics:
  - Average delivery ratio
  - Average latency
  - Average delivery overhead
- Experimental Wireless Sensor Testbed

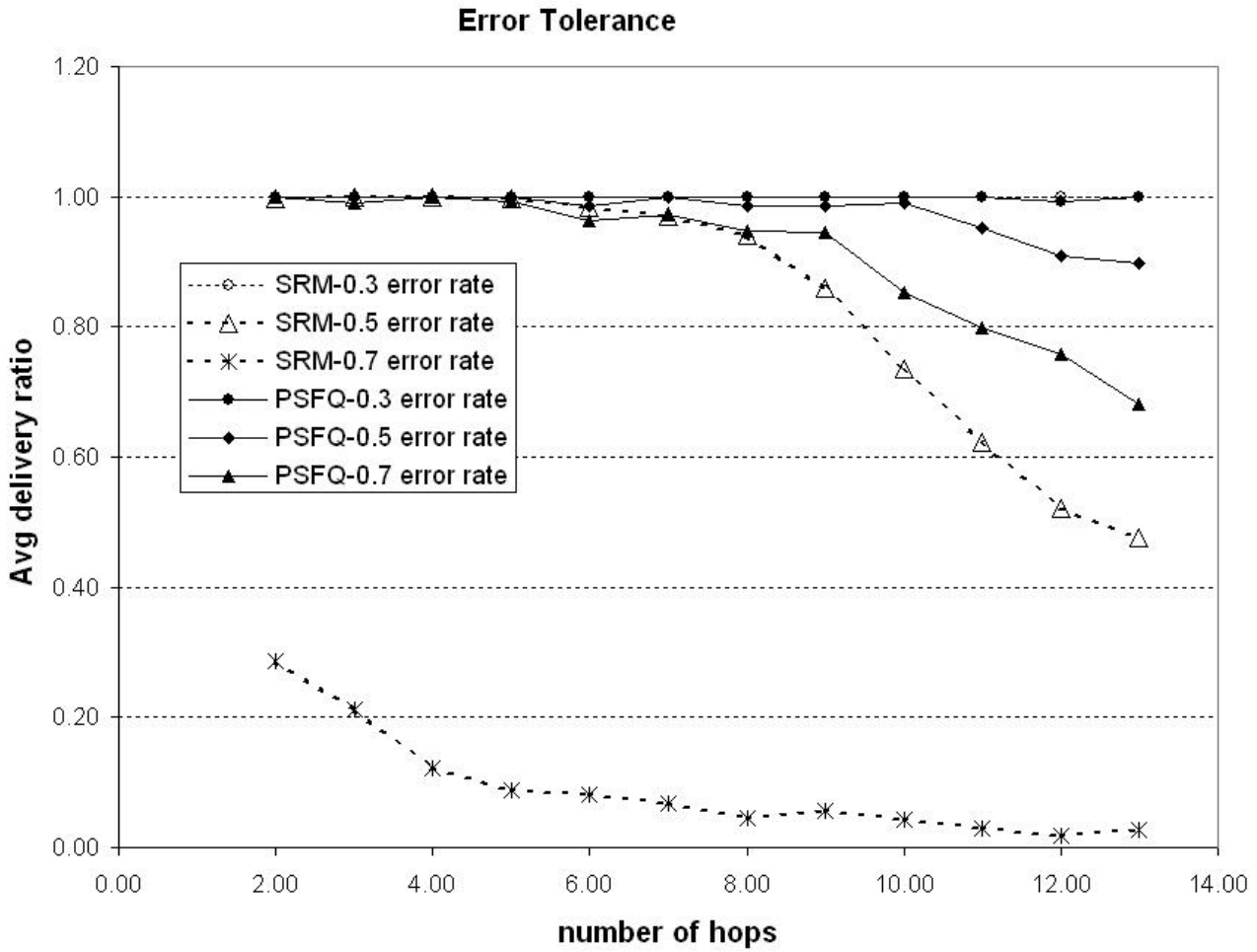
# Performance Evaluation



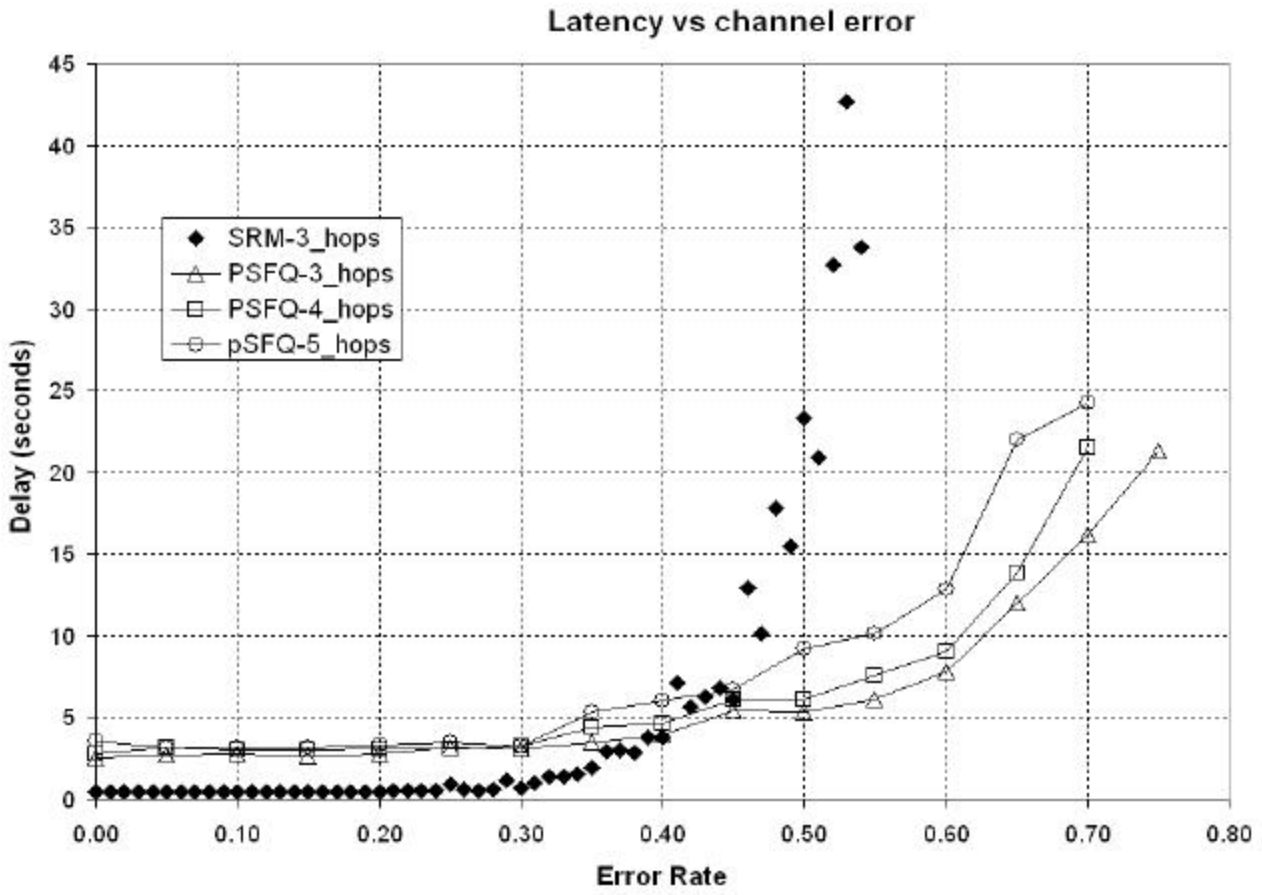

2Mbps, CSMA/CA channel access.

$T_{max}$ is 100ms, $T_{min}$ is 50ms and $T_r$ is 20ms

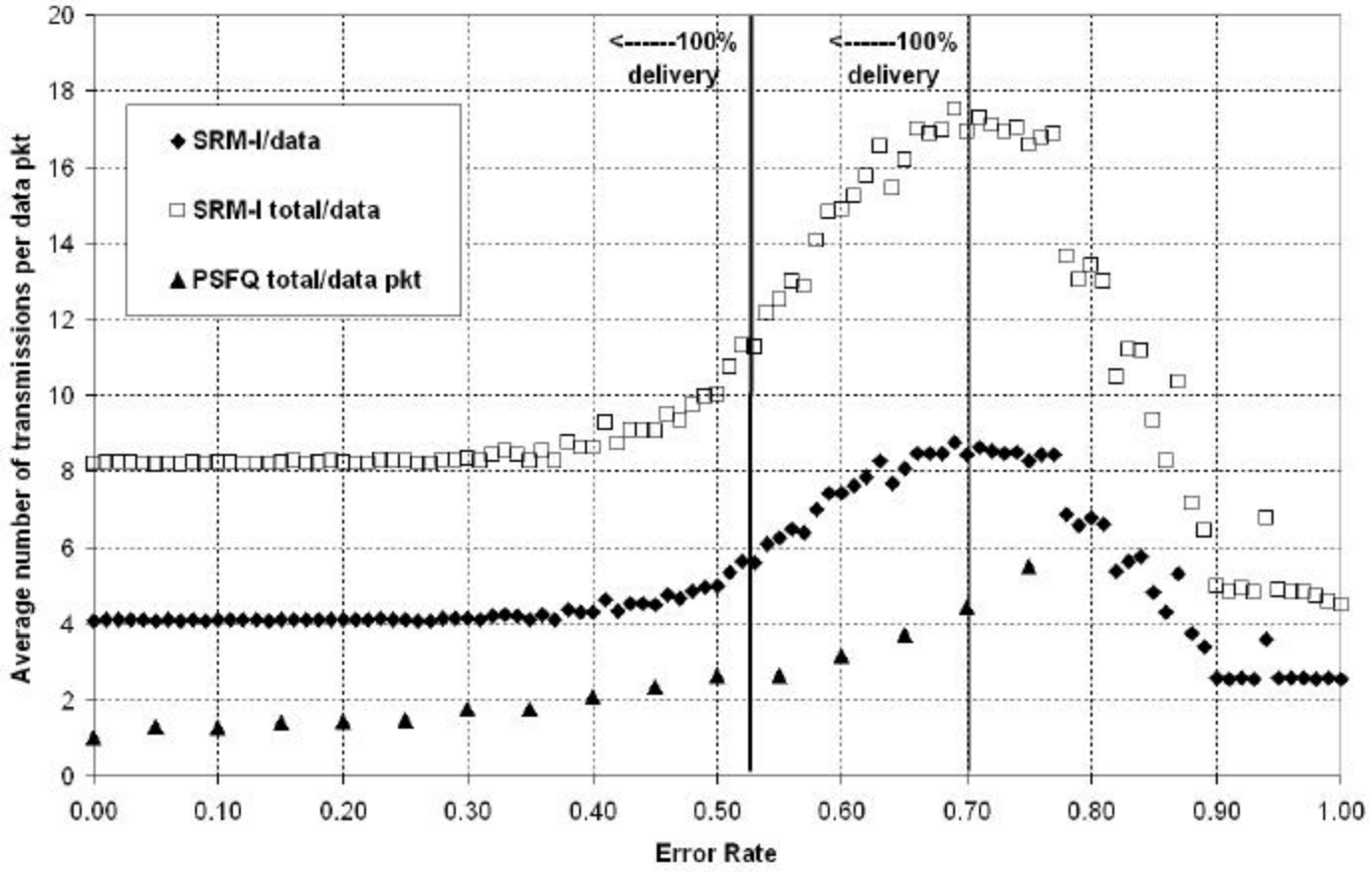
# Error Tolerance



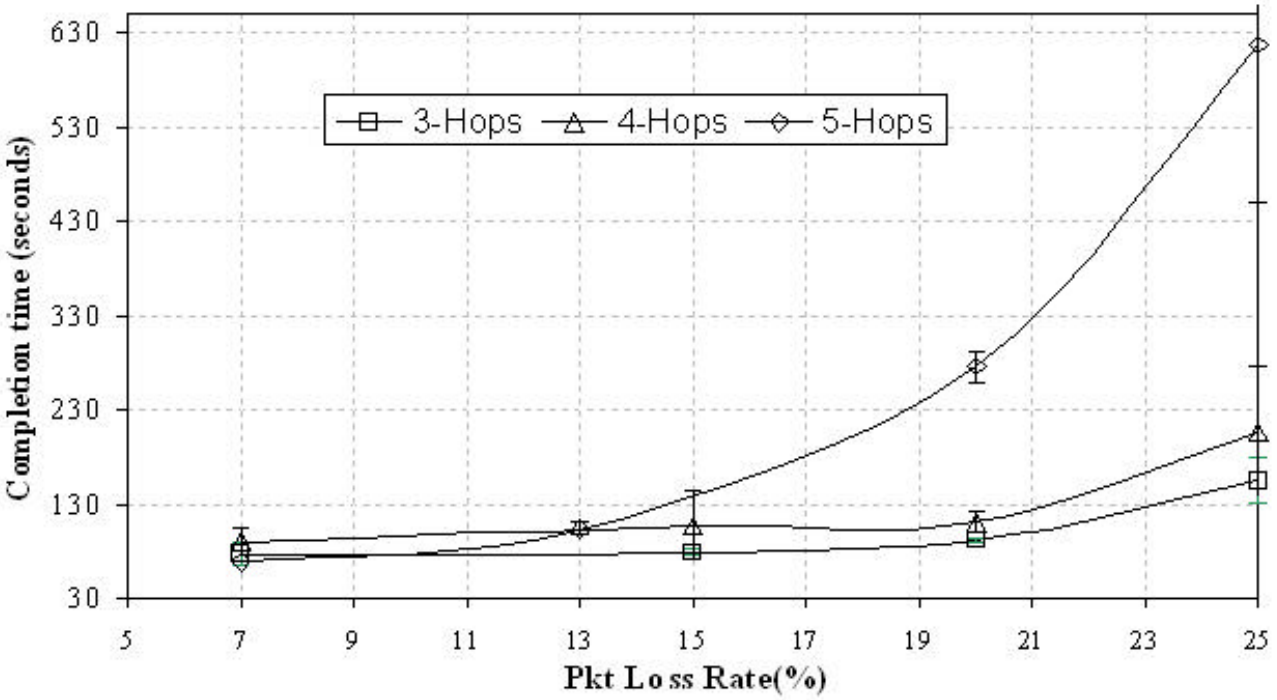
# Average Latency



# Communication Cost for Reliability



# Wireless Sensor Testbed




- $T_{max}$ = 0.3s and $T_r$ = 0.1s.

# Conclusion

- New reliable delivery scheme – showed proof-of-concept on real testbed.
- Large scale testbed experiment needed.
- Component source code release for TinyOS.