Online Appendix to: A Causal Approach to the Study of TCP Performance

HADRIEN HOURS, ERNST BIERSACK, and PATRICK LOISEAU, EURECOM

A. STUDY OF Z-FISHER AND KCI CRITERIA

In this appendix, we present the different tests that were executed to compare the performance of the KCI and Z-Fisher criteria. We generate artificial data with different functions and under different conditions.

A.1. Unconditional Tests

In these experiments, we generate independently two parameters, X and Y, and test, for different scenarios, the independences with the Z-fisher criterion and the KCI test. The results are summarized in Figure 7, where the different scenarios are tested 10 times for different sample sizes and the percentage of correct answers are presented. The variables are generated as follows:

—**Test 1** [Normal distribution, same variance]: The values for both parameters are drawn independently from two normal distributions $\mathcal{N}(0, 1)$
—**Test 2** [Normal distribution, different variance]: The values for both parameters are drawn independently from two different normal distributions, $X \sim \mathcal{N}(100, 5)$, $Y \sim \mathcal{N}(5, 2)$
—**Test 3** [Real parameter distribution, same variance]: We randomly draw samples from the RTT and throughput data values of Table II, and normalize them[10]
—**Test 4** [Real parameter normal distribution, different variance]: We randomly draw samples from the RTT and throughput data values of Table II

These results do not show any preference for one criteria over the other in any of the situations. They also do not show a clear improvement when the sample size increases.

A.2. Conditional Tests

In this section, we compare the independence criteria (Z-Fisher and Hilbert Schmidt Independence Criterion) in the case of conditional independence tests. We restrict ourselves to a conditional set of size 1. We have two possible configurations with three variables **X**, **Y**, and **Z**. In the first configuration, we have $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$, and in the second one, we have $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ and $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}$. For each configuration, several graphical representations, corresponding to the mechanisms generating these independences, exist. Some are presented Figure 8. However, the situations presented in Figures 8(b) and 8(c) are similar, so we will not differentiate them (inverting the role of **X** and **Y** will not give more information on the performance of a criteria). We will only study the cases represented by Figures 8(a), 8(b), and 8(d).

These structures of three variables are very important in causal model inference, as they represent the V-structures allowing the PC algorithm to start orienting some edges, which, in turn, will induce other orientations.

[10] $X_{normalized} = \frac{X - \mu_X}{\sigma_X}$.

DOI: http://dx.doi.org/10.1145/2770878

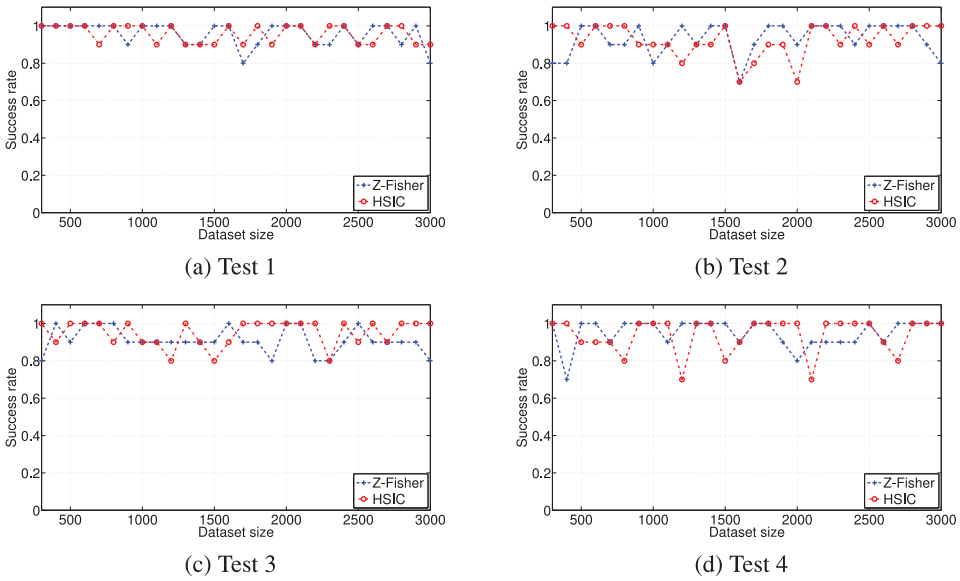


Fig. 7. Success rate of unconditional independence tests for the Z-Fisher criterion and KCI test, for the different cases and dataset sizes: Test 1: Normally distributed and same variance, Test 2: Normally distributed and different variance, Test 3: Not normally distributed and same variance, and Test 4: Not normally distributed and different variance.

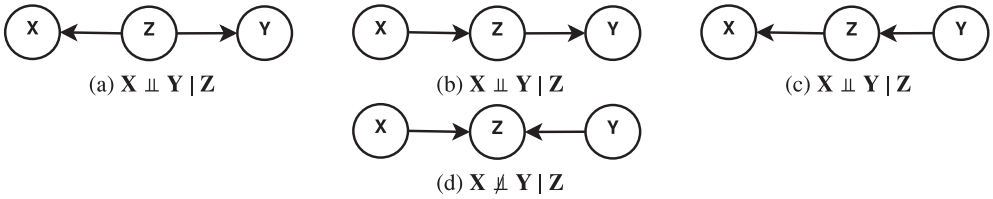


Fig. 8. Different graphical configurations for orienting V-structure.

For each case, there several possibilities:

—**Distribution:** normal/not normal
—**Dependences:** linear/not linear
—**Variance:** proportional/disproportional
—**Error terms:** presence/absence

These parameters can have an influence on the outcome of the independence test and will correspond to one series of tests for each case. For each of the three configurations, we run 16 tests, and for each test, different input sizes are used. The different tests are presented in Tables III-V and their corresponding results in Tables VI through VIII.

For the normally distributed case, we draw samples from a normal distribution, and for the non-normal case, we draw samples from the observations of the RTT and throughput from the real-case scenario dataset (Table II). As previously, for the normally distributed case, we select mean and variance according to the case we are testing. For the non-normal cases, if we want similar variance, we normalize the samples, and if we want different variance, we do not normalize the samples from RTT and throughput.

The first observation is that, when the dependences are deterministic, both criteria, rightfully, fail. The second observation is that the KCI has a success rate close to 80% or more over all the scenarios. On the other hand, the Z-Fisher criterion correctly detects

Table III. 16 Scenarios for Testing Conditional Independences with the Z-Fisher Criterion and KCI test, for the Case Illustrated in Figure 8(a), $Z^{\circ} = Z - min(Z) + 1$

Test	**Z**	**Functions**	$\sigma_{\mathbf{Z}}$	Error Terms
1	$Z \sim \mathcal{N}(0,1)$	$X = 5 \cdot Z$ $Y = -3 \cdot Z$	proportional	None
2	$Z \sim \mathcal{N}(0,1)$	$X = 5 \cdot Z$ $Y = -3 \cdot Z$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
3	$Z \sim \mathcal{N}(0,1)$	$X = 2 \cdot Z$ $Y = -300 \cdot Z$	disproportional	None
4	$Z \sim \mathcal{N}(0,1)$	$X = 2 \cdot Z$ $Y = -300 \cdot Z$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
5	$Z \sim \mathcal{N}(0,1)$	$X = \sqrt{5 \cdot Z^{\circ}}$ $Y = -3 \cdot \sqrt{Z^{\circ}}$	proportional	None
6	$Z \sim \mathcal{N}(0,1)$	$X = \sqrt{5 \cdot Z^{\circ}}$ $Y = -3 \cdot \sqrt{Z^{\circ}}$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
7	$Z \sim \mathcal{N}(0,1)$	$X = 3 \cdot \sqrt{500 \cdot Z^{\circ}}$ $Y = -2 \cdot \sqrt{3 \cdot Z^{\circ}}$	disproportional	None
8	$Z \sim \mathcal{N}(0,1)$	$X = 3 \cdot \sqrt{500 \cdot Z^{\circ}}$ $Y = -2 \cdot \sqrt{3 \cdot Z^{\circ}}$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
9	Z = not normal	$X = 5 \cdot Z$ $Y = -3 \cdot Z$	proportional	None
10	Z = not normal	$X = 5 \cdot Z$ $Y = -3 \cdot Z$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
11	Z = not normal	$X = 2 \cdot Z$ $Y = -300 \cdot Z$	disproportional	None
12	Z = not normal	$X = 2 \cdot Z$ $Y = -300 \cdot Z$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
13	Z = not normal	$X = \sqrt{5 \cdot Z^{\circ}}$ $Y = -3 \cdot \sqrt{Z^{\circ}}$	proportional	None
14	Z = not normal	$X = \sqrt{5 \cdot Z^{\circ}}$ $Y = -3 \cdot \sqrt{Z^{\circ}}$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
15	Z = not normal	$X = 3 \cdot \sqrt{500 \cdot Z^{\circ}}$ $Y = -2 \cdot \sqrt{3 \cdot Z^{\circ}}$	disproportional	None
16	Z = not normal	$X = 3 \cdot \sqrt{500 \cdot Z^{\circ}}$ $Y = -2 \cdot \sqrt{3 \cdot Z^{\circ}}$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$

only the dependence in the cases {2,4,10,12}, where the relationship between **X** and **Z** and **Y** and **Z** is linear.

As a conclusion, the KCI test always correctly detects the independences and dependencies. The KCI test does not seem to be impacted by the distribution of the variables or by the nature of their dependencies. While Fisher does not seem to difficulties with data that are not normally distributed, the absence of linearity makes it fail on every test of conditional independence. This observation led us to the conclusion that the Fisher test is not appropriate for our data.

Table IV. 16 Scenarios for Testing Conditional Independences with the Z-Fisher Criterion and KCI Test, for the Case Illustrated in Figure 8(b), $X^{\circ} = X - min(X) + 1$, $Y^{\circ} = Y - min(Y) + 1$

Test	**Z**	**Functions**	$\sigma_{\mathbf{Z}}$	Error Terms
1	$X \sim \mathcal{N}(0,1)$	$Y = 5 \cdot X$ $Z = -3 \cdot Y$	proportional	None
2	$X \sim \mathcal{N}(0,1)$	$Y = 5 \cdot X$ $Z = -3 \cdot Y$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
3	$X \sim \mathcal{N}(0,1)$	$Y = 2 \cdot X$ $Z = -300 \cdot Y$	disproportional	None
4	$X \sim \mathcal{N}(0,1)$	$Y = 2 \cdot X$ $Z = -300 \cdot Y$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
5	$X \sim \mathcal{N}(0,1)$	$Y = \sqrt{5 \cdot X^{\circ}}$ $Z = -3 \cot \sqrt{Y^{\circ}}$	proportional	None
6	$X \sim \mathcal{N}(0,1)$	$Y = \sqrt{5 \cdot X^{\circ}}$ $Z = -3 \cot \sqrt{Y^{\circ}}$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
7	$X \sim \mathcal{N}(0,1)$	$Y = 3 \cot \sqrt{500 \cdot X^{\circ}}$ $Z = -2 \cdot \sqrt{3 \cdot Y^{\circ}}$	disproportional	None
8	$X \sim \mathcal{N}(0,1)$	$Y = 3 \cot \sqrt{500 \cdot X^{\circ *}}$ $Z = -2 \cdot \sqrt{3 \cdot Y^{\circ}}$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
9	X = not normal	$Y = 5 \cdot X$ $Z = -3 \cdot Y$	proportional	None
10	X = not normal	$Y = 5 \cdot X$ $Z = -3 \cdot Y$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
11	X = not normal	$Y = 2 \cdot X$ $Z = -300 \cdot Y$	disproportional	None
12	X = not normal	$Y = 2 \cdot X$ $Z = -300 \cdot Y$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
13	X = not normal	$Y = \sqrt{5 \cdot X^{\circ}}$ $Z = -3 \cot \sqrt{Y^{\circ}}$	proportional	None
14	X = not normal	$Y = \sqrt{5 \cdot X^{\circ}}$ $Z = -3 \cot \sqrt{Y^{\circ}}$	proportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$
15	X = not normal	$Y = 3 \cot \sqrt{500 \cdot X^{\circ}}$ $Z = -2 \cdot \sqrt{3 \cdot Y^{\circ}}$	disproportional	None
16	X = not normal	$Y = 3 \cot \sqrt{500 \cdot X^{\circ}}$ $Z = -2 \cdot \sqrt{3 \cdot Y^{\circ}}$	disproportional	$\varepsilon_{X/Y} \sim \mathcal{N}(0, 0.1)$

Table V. 16 Scenarios for Testing Conditional Independences with the Z-Fisher Criterion and KCI Test, for the Case Illustrated in Figure 8(d), $X^{\circ} = X - min(X) + 1$

Test	**Z**	**Functions**	$\sigma_{\mathbf{Z}}$	Error Terms
1	$X \sim \mathcal{N}(0,1)$ $Y \sim \mathcal{N}(0,1)$	$Z = 5 * X - 3 * Y$	proportional	None
2	$X \sim \mathcal{N}(0,1)$ $Y \sim \mathcal{N}(0,1)$	$Z = 5 * X - 3 * Y$	proportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
3	$X \sim \mathcal{N}(5,10)$ $Y \sim \mathcal{N}(20,100)$	$Z = 5 * X - 3 * Y$	disproportional	None
4	$X \sim \mathcal{N}(5,10)$ $Y \sim \mathcal{N}(20,100)$	$Z = 5 * X - 3 * Y$	disproportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
5	$X \sim \mathcal{N}(0,1)$ $Y \sim \mathcal{N}(0,1)$	$Z = -3 * \sqrt{(X+Y)^{\circ}}$	proportional	None
6	$X \sim \mathcal{N}(0,1)$ $Y \sim \mathcal{N}(0,1)$	$Z = -3 * \sqrt{(X+Y)^{\circ}}$	proportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
7	$X \sim \mathcal{N}(5,10)$ $Y \sim \mathcal{N}(20,100)$	$Z = -3 * \sqrt{(X+Y)^{\circ}}$	disproportional	None
8	$X \sim \mathcal{N}(5,10)$ $Y \sim \mathcal{N}(20,100)$	$Z = -3 * \sqrt{(X+Y)^{\circ}}$	disproportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
9	X = not normal Y = not normal	$Z = 5 * X - 3 * Y$	proportional	None
10	X = not normal Y = not normal	$Z = 5 * X - 3 * Y$	proportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
11	X = not normal Y = not normal	$Z = -5 * X - 300 * Y$	disproportional	None
12	X = not normal Y = not normal	$Z = 5 * X - 300 * Y$	disproportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
13	X = not normal Y = not normal	$Z = -3 * \sqrt{(X+Y)^{\circ}}$	proportional	None
14	X = not normal Y = not normal	$Z = -3 * \sqrt{(X+Y)^{\circ}}$	proportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$
15	X = not normal Y = not normal	$Z = -2 * \sqrt{3 * (5 * X + 300 * Y)^{\circ}}$	disproportional	None
16	X = not normal Y = not normal	$Z = -2 * \sqrt{3 * (5 * X + 300 * Y)^{\circ}}$	disproportional	$\varepsilon_Z \sim \mathcal{N}(0, 0.1)$

Table VI. Results of the 16 Conditional Independence Scenarios for the Model in Figure 8(a), Tested Using the Z-Fisher Criterion and KCI Test for 6 Different Dataset Sizes, Averaged on 10 Trials, $\mathcal{I}_1 = X \perp\!\!\!\perp Y, \mathcal{I}_2 = X \perp\!\!\!\perp Y \mid Z$

Test	500				1,000				1,500				2,000				2,500				3,000			
	$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$	
	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H
1	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0.1	0
2	0	0	0.8	1	0	0	1	1	0	0	0.9	0.9	0	0	1	0.8	0	0	0.8	0.9	0	0	0.9	0.8
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	0	0	0.8	0.9	0	0	1	1	0	0	1	1	0	0	0.9	1	0	0	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	1	0	0	0	0.9	0	0	0	0.9	0	0	0	0.9	0	0	0	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0.9	0	0	0	1	0	0	0	0.8	0	0	0	0.8	0	0	0	0.9	0	0	0	1
9	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	1	1	0	0	1	0.9	0	0	1	0.8	0	0	0.9	0.8	0	0	0.9	1	0	0	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0.9	0.9	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0.8	1	0	0	0.9	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0.7	0	0	0	0.8	0	0	0	0.6	0	0	0	0.7	0	0	0	0.9	0	0	0	0.9
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	1	0	0	0	0.8	0	0	0	1	0	0	0	0.9	0	0	0	1	0	0	0	1

Table VII. Results of the 16 Conditional Independence Scenarios for the Model in Figure 8(b) for the Z-Fisher Criterion and KCI Test, for 6 Different Dataset Sizes, Averaged on 10 Trials, $\mathcal{I}_1 = X \perp\!\!\!\perp Y$, $\mathcal{I}_2 = X \perp\!\!\!\perp Y \mid Z$

Test	500				1,000				1,500				2,000				2,500				3,000			
	$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$	
	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H
1	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0.9	0	0	1	1	0	0	0.9	0.9	0	0	1	0.9	0	0	0.9	1	0	0	1	0.9
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	0	0	0.8	0.8	0	0	0.8	1	0	0	0.8	1	0	0	0.9	0.9	0	0	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	1	0	0	0.1	1	0	0	0	1	0	0	0	1	0	0	0	1
7	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0.9	0	0	0	1	0	0	0	0.9	0	0	0	1	0	0	0	0.9
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0.8	1	0	0	1	1	0	0	1	0.9
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	1	1	0	0	0.9	0.9	0	0	1	1	0	0	1	0.9	0	0	1	1	0	0	1	0.9
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0.9	0	0	0	1	0	0	0	1	0	0	0	0.9	0	0	0	1	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0.9	0	0	0	1	0	0	0	1	0	0	0	0.9	0	0	0	0.9	0	0	0	1

Table VIII. Results of the 16 Tests for the Model Figure 8(d) for 6 Different Dataset Sizes Using the Fisher Criterion and KCI Test $\mathcal{I}_1 = X \perp\!\!\!\perp Y$, $\mathcal{I}_2 = X \perp\!\!\!\perp Y/Z$.

Test	500				1,000				1,500				2,000				2,500				3,000			
	$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$		$\mathcal{I}_1$		$\mathcal{I}_2$	
	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H	F	H
1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
2	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
3	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
4	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
5	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
6	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
7	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
8	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
9	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
10	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
11	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
12	1	1	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	1	0	0
13	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0
14	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0

B. INDEPENDENCE TEST: KCI AND BOOTSTRAP

In this appendix, we present experiments on the efficiency and accuracy of the KCI tests (with and without our bootstrap method) that support our choice of using the bootstrap and our choices of parameters presented in Section 3.1.3 (l and N).

B.1. KCI Test Algorithm and Numerical Considerations

We first briefly describes the KCI test principle and the important numerical considerations that affect its accuracy.

The Kernel-based Conditional Independence test (KCI) maps the initial dimensions into a new space, Hilbert Schmidt Space (HSS), using definite positive kernels. The translation can be seen as a bijective operation. In the new space, Reproducing Kernel Hilbert Space (RKHS), a new operator, the cross-covariance operator, inspired by the covariance and conditional covariance in the original spaces, allows to test the non-linear independences. The computation of the cross-covariance operator and conditional cross-covariance operator can be approximated and some simplifications allows to reduce the complexity of the independence test between two parameters. However, one of these simplifications makes use of the Cholesky factorization, which tends to fail if the variability of one of tested parameter is not big enough. This is a known numerical limitation of the HSIC implementation (see, e.g., Appendix B in Tipping [2001]); due to a matrix, which, although theoretically invertible, appears singular in practice.

Taking as example the conditional test of the *time of the day* and *distance* conditionally on {*bufferingdelay*,*rtt*,*p*} for the dataset presented in Table II, we observed that the KCI test on the full dataset fails 100% of the time, in its use of Cholesky factorization. This problem was initially solved by rejecting the H_0 hypothesis in case of failure, but this solution is not satisfying. In contrast, the bootstrap method overcomes this limitation for two reasons: (i) using smaller re-sampled datasets ($N = 400$ for us) reduces the numerical problems in the Cholesky factorization, and (ii) out of a large number ($l = 100$ for us) of randomly re-sampled datasets, we always find many where the Cholesky factorization does not fail. As a result, we are able to detect independence in datasets where the Cholesky factorization was systematically failing with the KCI test alone on the full dataset, which, to some extent, can be interpreted as an increase of the test accuracy thanks to the bootstrap method.

B.2. Completion Time

We now discuss computation time constraints.

B.2.1. KCI Test Completion Time. In a first experiment, we record the time it takes for the KCI test implementation from Zhang et al. [2012] to complete an independence test for different dataset sizes and for different conditional set sizes.[11]

The results are shown in Figure 9. All graphs show that the completion time seems to increase faster than linearly with the number of samples. In practice, for a sample size larger than 400, the completion time starts to increase very fast with the number of samples. Together with the observation that no important improvements are obtained for sample sizes bigger than 400 samples (see Section B.3), this motivated our choice of the value 400 for N.

B.2.2. Influence of Bootstrap Parameterization on Completion Time. We now provide estimates of real computation times to infer a graph with the bootstrap method for different

[11]Note that this completion time was obtained with the version 2009b of the MATLAB software, while the same trends are still valid for latest release of MATLAB.

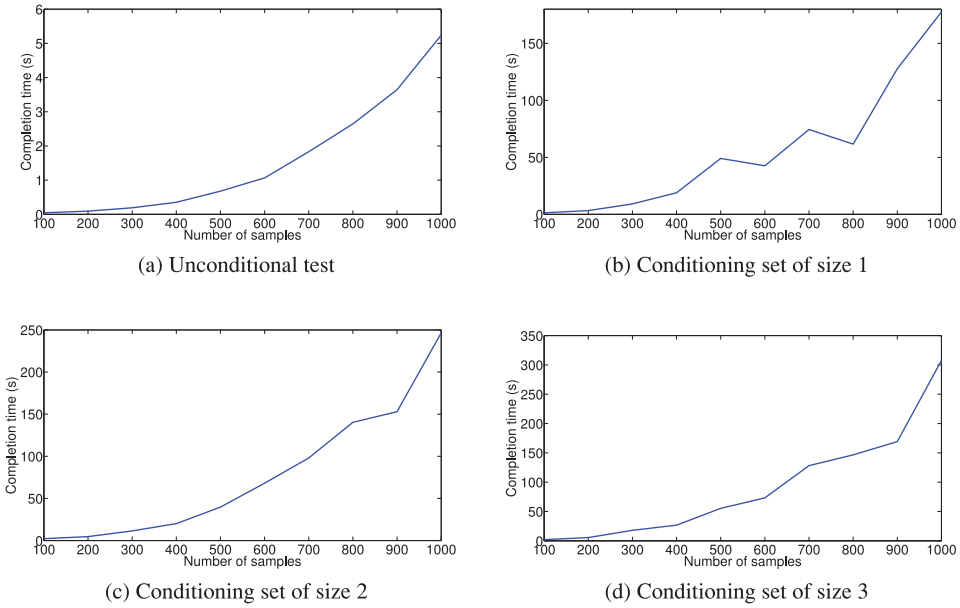


Fig. 9. Evolution of the completion time of the KCI test as function of the number of samples for different conditioning set sizes.

parameters. Taking as an example the dataset from Section 5 and the number of independences tested by the PC algorithm, Table IX presents the time it would take for different values of our bootstrap implementation to complete the inference of the causal graph. The values were obtained by counting the number of independences for each conditioning set size in the case where the Z-Fisher criterion is used and by recording the time it takes for each type of independence test with different parameterization in our bootstrap method.

Note that, due to time constraints, the different estimations had to be run in parallel on different machines with slightly different performances: 12 CPUs, 2,260MHz, 30GB or 8 CPUs, 1,862MHz, 15GB (but note that only 1 or 2 CPU was used during the tests, so the difference in the number of CPUs is not important). This explains small variations in the estimates (e.g., that the completion time estimate for $\{N = 200, l = 1{,}000\}$ is found smaller than for $\{N = 100, l = 1{,}000\}$), but this does not affect the main trends.

Table IX highlights important time constraints that limit the number of sub-tests in the bootstrap. Indeed, for $N = 400$, computation times are of the order of days and go to almost a year for $l = 1{,}000$. It should be noted, however, that one very important advantage of the bootstrap method (in addition to detecting independences in cases where the basic KCI test would fail, see Section B.1), is the possibility it offers to parallelize the computation very easily. Indeed, when using the bootstrap method, each test on a re-sampled dataset can simply be performed on a different machine. By using M machines, we can then shrink the completion time by a factor proportional to the number of machines used. As it is now very common for a university or company to work with a cluster of machines (virtually or not), this is a very important advantage. For instance, at our university, we have around $M = 50$ machines. For $N = 400$, this reduces the computation time from 33 days to 18 hours. However, this still limits the number of re-sampling. For instance, for $N = 400$, using $l = 1{,}000$, the algorithm would still take 5 days. While 5 days could still seem an acceptable time (although it can become unmanageable with less resources or if tests need to be run several times), the important point is that it corresponds to a very large increase compared to

Table IX. Impact of Bootstrap Parameters on Completion Time

N	l	time (hours)	time (days)
100	1	2.1	0.09
200	1	2.5	0.1
400	1	2.9	0.12
600	1	23.9	1.0
800	1	44.0	1.83
1,000	1	69.0	2.87
100	10	5.4	0.22
200	10	19.5	0.81
400	10	45.5	1.9
600	10	201.6	8.4
800	10	427.2	18.7
1,000	10	839.1	37.18
100	100	52.3	2.18
200	100	283.6	11.82
400	100	804.3	33.51
600	100	924.8	38.53
800	100	4,647.1	193.63
1,000	100	9,688.6	403.69
100	1,000	2,857.4	119.06
200	1,000	2,108.8	87.87
400	1,000	5,935.6	247.31
600	1,000	15,657.8	652.40
800	1,000	30,503.5	1,270.97
1,000	1,000	31,796.9	1,324.86

18 hours. In the model used for this estimation, we consider only 12 parameters, but this difference will rapidly become difficult to handle as soon as more complex systems, with more parameters and independences to test, are studied.

B.3. KCI and Bootstrap Tests Accuracy

We finally compare the accuracy of the KCI test with and without bootstrap, in situations close to the ones we have to deal with in our system.

The notion of performance/accuracy of the test can only be discussed in cases where the independences are known a priori. To estimate the type I and II errors of the test in a conditional independence setting, we consider the graph of Figure 11. In order to stay close to our original system, we generate X_1 by randomly re-sampling with replacement the throughput, from the dataset of Table II first, and from the dataset of Table I after, to keep the same distribution; and we generate X_2, X_3 and X_4 using non-linear functions:

$X_1 = \text{re-sampling(tput)};$
$X_2 = \sqrt{10X_1} + \mathcal{N}(0, 0.8)$
$X_3 = -5\sqrt{0.5X_1} + \mathcal{N}(0, 0.8)$
$X_4 = \sqrt{6X_2 - 2X_3} + \mathcal{N}(0, 0.8)$

To estimate both types of errors, we test the following independences:

—$\mathcal{I}_\mathbf{1}$: $X_1 \perp\!\!\!\perp X_4 | \{X_2, X_3\}$: H_0 should be accepted;
—$\mathcal{I}_\mathbf{2}$: $X_2 \perp\!\!\!\perp X_3 | \{X_1, X_4\}$: H_0 should be rejected.

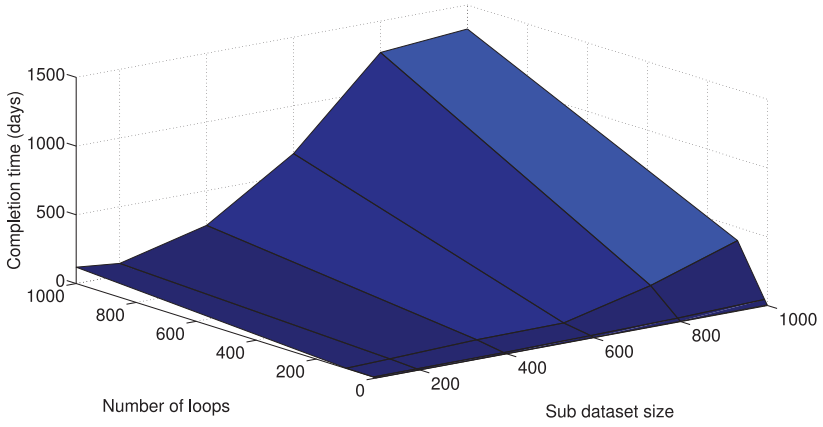


Fig. 10. Evolution of the completion time for different values of the bootstrap method.

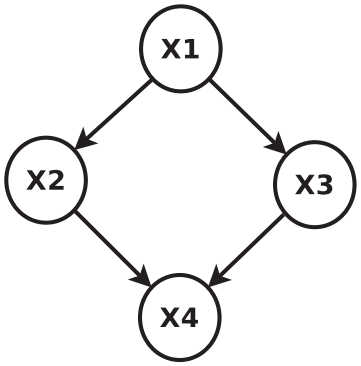
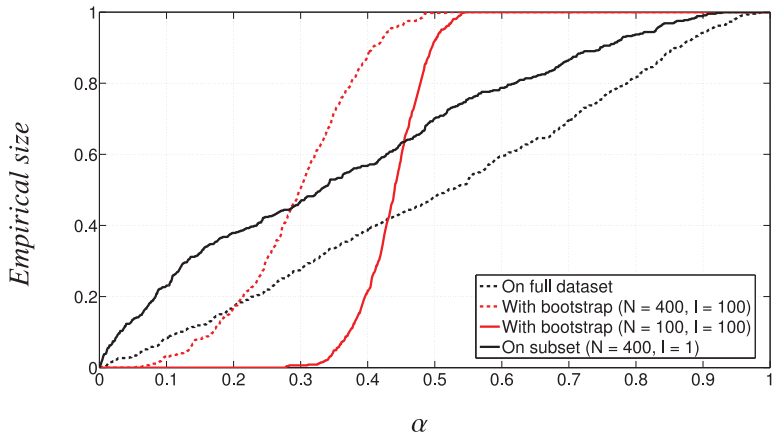


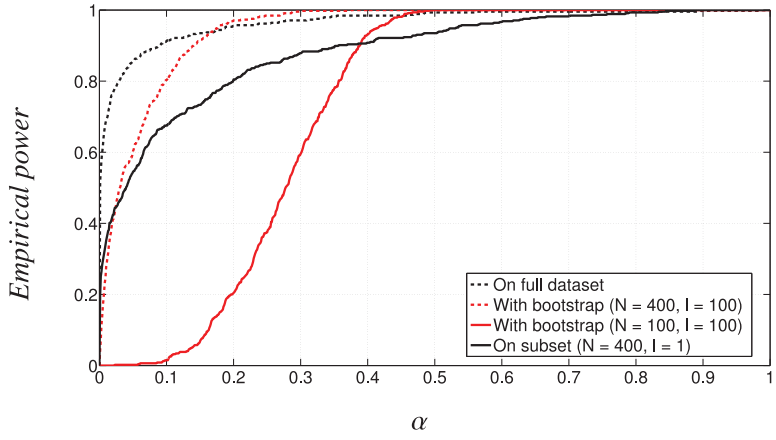
Fig. 11. Causal graph of the artificial dataset of four variables.

B.3.1. Generating X1 from the Throughput Observed at the FTP Server in the Dataset of Table II in Section 5. Figures 12(a) and 12(b) present the results in terms of size and power of the test as a function of the parameter α (recall that a test rejects H_0 if the p-value is smaller than α). The result for the KCI test on the full dataset is shown with the black dotted line, as the baseline curve. To assess the accuracy of the test using the bootstrap method on smaller re-sampled datasets, we compare two sets of parameters ($N = 400$, $l = 100$), the red dotted line, and ($N = 100$, $l = 100$), the red solid line. For comparison and interpretation, we also include the results for ($N = 400$, $l = 1$), the black solid line, which corresponds to a method doing the test on one re-sampled dataset of smaller size rather than on the full dataset (almost equivalent to taking a subset except that the re-sampling is done with replacement).

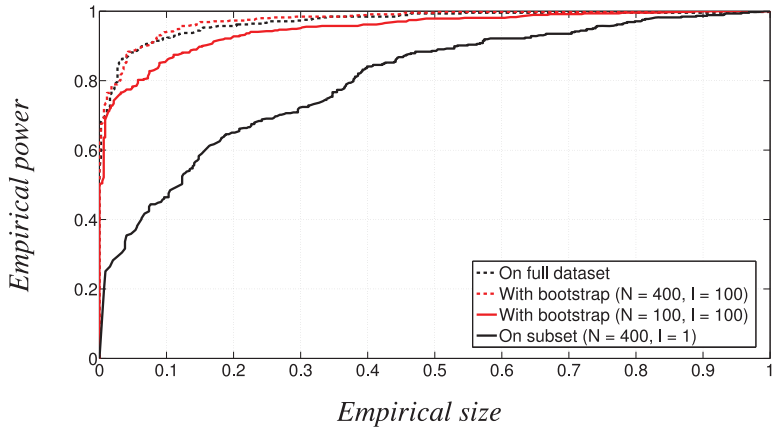
The size, defined as the probability of rejecting H_0 under H_0 (probability of type I error), is estimated by testing independence $\mathcal{I}_1$ on 446 datasets generated as described above (each with 1,000 samples). Figure 12(a) shows that, as expected, the size for the full dataset test and for the test on a subset are close to the first bi-sector ($size = \alpha$). For the tests with bootstrap, however, the sizes are very different from α. This is normal since the test can no longer be simply defined as comparing a p-value to α. The power, defined as the probability of rejecting H_0 under H_1 (complementary of the probability of type II error) is estimated by testing independence $\mathcal{I}_2$ on 446 datasets generated as described above (each with 1,000 samples). Figure 12(b) shows that, naturally, for a given α, the power for the test on a subset (black solid line) is smaller than the power



(a) Empirical test size (i.e., fraction of incorrectly rejected independence $\mathcal{I}_1$) as function of test parameter α.



(b) Empirical test power (i.e., fraction of correctly rejected independence $\mathcal{I}_2$) as function of test parameter α.



(c) Empirical test power as a function of the empirical test size.

Fig. 12. Comparison of size and power of the KCI test with and without the use of bootstrap.

for the test on the full dataset (black dotted line). The comparison of powers with tests using bootstrap is not meaningful, however, because, for a given α, the sizes are very different.

Instead, Figure 12(c) shows the real accuracy trade-off between size and power achieved by each test. The most important result is that the test with bootstrap with parameters $N = 400$ and $l = 100$ (red dotted line) achieves the same performance as the KCI test on the full dataset (black dotted line). This validates, in a situation very close to the one encountered in our system (conditional independences, non-normal distributions and non-linear dependences), both the bootstrap procedure and the choice of parameters. As a comparison, the test on a subset of 400 samples achieves a very poor performance (which shows the importance of the bootstrap) and the test with parameters $N = 100$ and $l = 100$ achieves a smaller but still reasonable performance (which shows that further reductions of the computation time are possible but not without a small degradation of the accuracy).

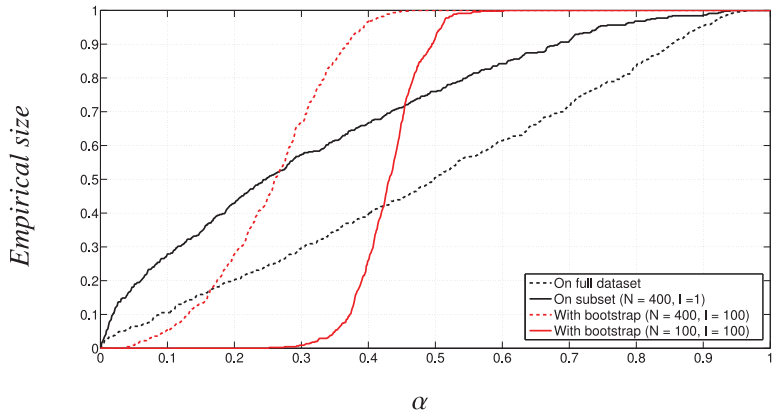
B.3.2. Generating X1 from the Throughput Observed at the FTP server in the Dataset of Table I in Section 4. We repeat the same experience as previously, but this time X_1 is generated from the throughput of the emulated network (Table I). The size, the power, and the power as a function of the size are represented in Figures 13(a), 13(b), and 13(c), respectively.

We focus our discussion on the graph of the power as function of the size (Figure 13(c)). We can observe that the performance of the KCI test when using $l = 100$ and $N = 400$ is not as close to the performance obtained with the full dataset as when generating X_1 from the real FTP traffic throughput (Figure 12(c)). However, we still observe less than 5% difference, which, for our work, we consider acceptable as compared to the loss in terms of resources that would occur if choosing a higher number of loops or a bigger dataset to increase accuracy. Again, the choice of N and l results from a trade-off between precision and resources, and, as in the previous section, the choice of $N = 400$ and $l = 100$ offers a trade-off that comply with our needs. An important point to notice here is that this setting was tested in the emulated network scenario (Section 4). The graphical causal model we obtain based on the sequence of independence tests using this setting ($N = 400$, $l = 100$) was used to predict interventions that could be then verified by manually intervening on the system which further validates our choice of parameters.

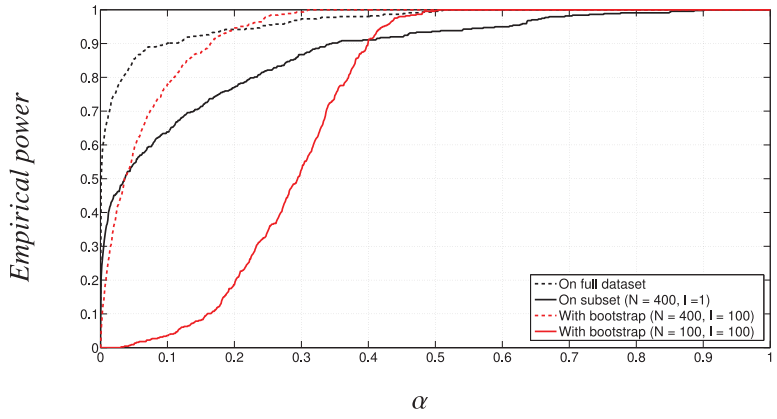
B.3.3. Concluding Remarks. The important difference between the previous two cases is that, in the case where X_1 was generated from the emulated network throughput, the setting $\{N = 100, S = 400\}$ was used to infer the graph that supported the predictions we made and that could be verified. As independences are not known in advance, we used two criteria to validate our setting:

(1) We generate an artificial dataset with known independences, distributions similar to the ones of the data we observe in our work and dependences inspired by the system we observe and our domain knowledge. We then use this ground truth to parameterize our algorithm.
(2) For a given parameterization, for a system where the dependences are not known, we use the obtained setting to infer a graph that is used to predict an intervention that can be performed and verified on the system after. This last point validates the different steps.

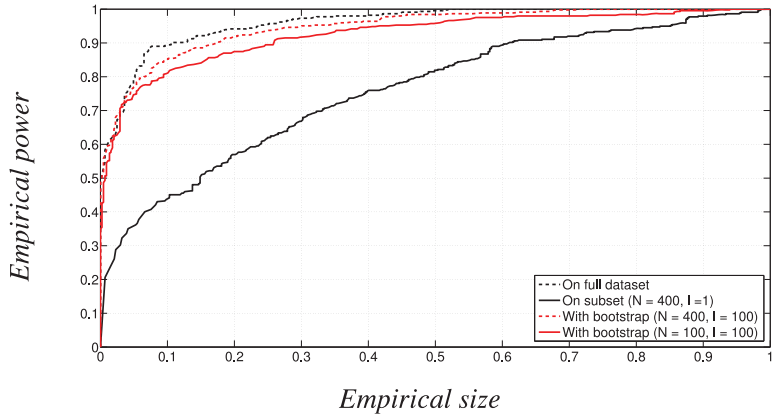
The fact that, for both scenarios, the setting $N = 400$ and $l = 100$ meets our objectives in terms of accuracy and resources validates this choice. The small difference in the observed performance of the test for the two scenarios is also validating the approach, as (i) we can see that the two datasets are different but the same setting can be



(a) Empirical test size (i.e., fraction of incorrectly rejected independence $\mathcal{I}_1$) as function of test parameter α.



(b) Empirical test power (i.e., fraction of correctly rejected independence $\mathcal{I}_2$) as function of test parameter α.



(c) Empirical test power as a function of the empirical test size.

Fig. 13. Comparison of size and power of the KCI test with and without the use of bootstrap for X_1 generated from the emulated network throughput.

applied, and (ii) the second scenario, where the performance is not as good as in the first one, corresponds to the case where the distribution of X_1 follows the distribution of a parameter of the emulated network where the predictions could be verified and the approach validated. This last point suggests that we can expect the performance of this setting of the KCI test + bootstrap to be at least as good in the real case scenario as in the emulated case, where we could verify that our methods perform correctly.

C. TCP PERFORMANCE PARAMETERS

C.1. List and Definition of the Different Parameters

In the emulated environment, we record the following parameters:

—*Bandwidth:* The maximum capacity of the path between the client and the server [*bw*]
—*Propagation Delay:* The time, when no queuing happens, for a packet to go from the server to the client [*delay*]
—*Size of Buffer:* The maximum number of packets that can be stored by a router, R_X, in case of congestion [*queueX*]
—*Narrow Link Available Capacity:* The estimated capacity of the path between the client and the server that the client has access to (takes into account cross traffic) [*buffersize*]
—*Receiver Window:* The client advertised receiver window, which captures the amount of packets that the client can process [*rwin*]
—*Buffering Delay:* The fraction of the time it takes for a packet to cross the network that is due to queuing in router buffers [*bufferingdelay*]
—*Round Trip Time:* The time it takes for a packet to cross the network (and be acknowledged) [*rtt*]
—*Timeouts*: The number of retransmissions triggered at the server for time out reason (no acknowledgement received) [*timeouts*]
—*Retransmission Score:* The fraction of packets that had to be retransmitted by the server (approximate the loss frequency) [*retrscore*]
—*Probability of a Loss Event:* The fraction of loss events, where a loss event is the occurrence of a burst of packets being lost (dropped by a router) [*p*]
—*Number of Bytes:* Total amount of bytes sent by the server to the client [*nbbytes*]
—*Throughput:* Amount of bytes that the server was able to send in a given amount of time to the client [*tput*]

In the real case scenario, due to the fact that we place ourselves at the edge of the network, we do not record parameters that describe the state of the routers. However, we additionally record the following parameters:

—*Distance:* Distance between the server and the client [*dist*]
—*Time of the Day:* Number of elapsed seconds since midnight when the connections was opened. This parameter captures the peak hour effect [*tod*]
—*Number of Hops:* Estimation of the number of routers between the server and the client

C.2. Empirical Distributions of Some Key Parameters

In this section, we present the histogram of some of the parameters of the real FTP traffic dataset, described in Table II in Section 5. Figure 14 shows the histograms of the following four parameters: *nbbytes*, *RTT*, *p*, and *throughput*. These histograms show that the parameters do not follow a specific distribution.

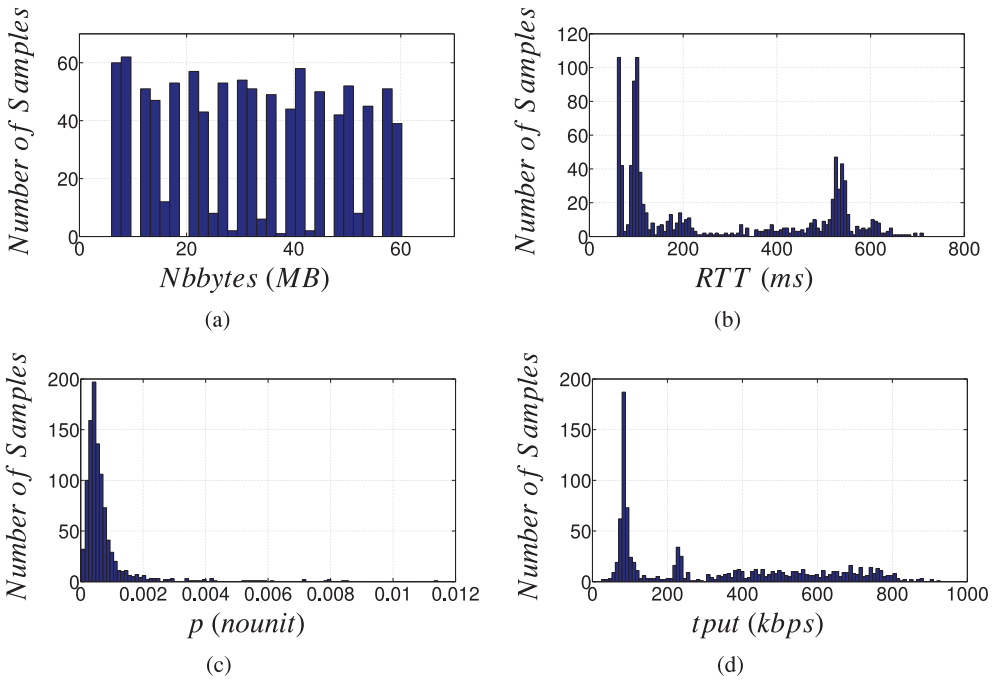


Fig. 14. Histograms of the *nbbytes (a)*, *rtt (b)*, *p (c),* and *tput (d)* parameters observed in the study of Real FTP traffic.

D. ADDITIONAL RESULTS FOR THE MININET DATASET

D.1. Causal Model

D.1.1. Randomizing the Simulation. We repeated the simulation presented in Section 4.2 by randomly selecting the values of the simulation parameters (delay, jitter, bandwidth, buffer size).

The corresponding dataset is presented in Table X and the corresponding causal model in Figure 15. We can observe that the graph is lacking many important edges when compared to the one obtained with no randomization (Figure 2 of Section 4.3). In the experiment we set up in Section 4.2, we choose the values of the parameters such that to create situations that exhibit some TCP properties due to network congestion or application limitations. Here, by randomly selecting the values of these parameters, these situations happen less often and the dependences are more difficult to detect. Due to time constraints, we could not test a bigger range of network scenarios. While many edges we found in the model presented Figure 2 are not present in Figure 15, we can still observe the parameters p, *rtt*, and *rwin* as direct parents of the throughput (*tput*). We also observe *delay* and *bufferingdelay* as direct parents of the *rtt*. Eventually, the dependence between the bandwidth (*bw*) and propagation delay (*delay*) is not present anymore.

Table X. Summary of the Randomly Emulated Network Dataset

Parameter	Definition	Min	Max	Avg	CoV
bw	minimum bandwidth (MBps)	1	25	6.7	0.77
delay	propagation delay (ms)	50	170	96	0.37
queue1	size of **R1** buffer (pkts)	10	400	98	1.1
queue2	size of **R2** buffer (pkts)	10	400	86	1.0
nlac	Narrow Link Available Capacity (kBps)	39	1.06e+5	1.15e+5	2.5
rwin	Receiver window advertised by **C1** (KB)	69	793	201	0.58
bufferingdelay	part of the RTT due to queuing delay (ms)	0.43	288	38.2	1.1
rtt	Round Trip Time (ms)	82	1074	221	0.67
timeouts	number of timeouts (units)	0	4	272	2.0
retrscore	fraction of retransmitted packets (no unit)	0	0.7	0.006	4.2
p	fraction of loss events (no unit)	0	0.38	0.004	5.1
nbbytes	number of bytes sent by the server (MB)	70	154	110	0.21
tput	throughput (kBps)	41	797	228	0.7

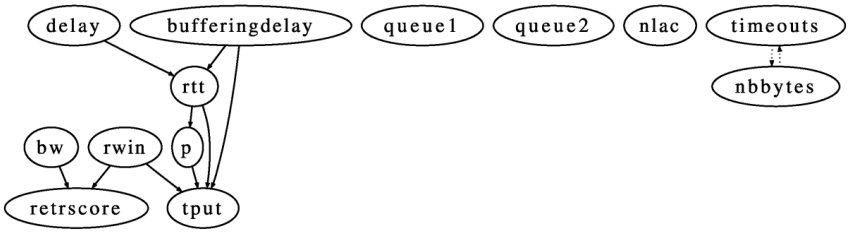


Fig. 15. Causal model obtained for the emulated network scenario when randomization is used.

D.1.2. Z-Fisher Criterion. Figure 16 represents the model inferred by the PC algorithm when we use the Z-Fisher criterion instead of our modified version of the KCI test.

We first notice that the model of Figure 16 is quite different from the one presented in Figure 2. The absence of *retrscore* as a parent of *tput* is an important dependence that is not present here. The path *bufferingdelay* $\rightarrow$ *bw* $\rightarrow$ *delay* $\rightarrow$ *queue2* $\rightarrow$ *queue1* $\rightarrow$ *timeouts* does not find any explanation from our domain knowledge of the TCP mechanisms and cannot be explained, as in the previous case, by the nature of experimental design. While the model inferred using the KCI test presents properties and dependencies that are supported by the domain knowledge of TCP performance, the model inferred with the Z-Fisher criterion does not (for an experimental comparison of the two criteria, we refer the reader to Appendix A).

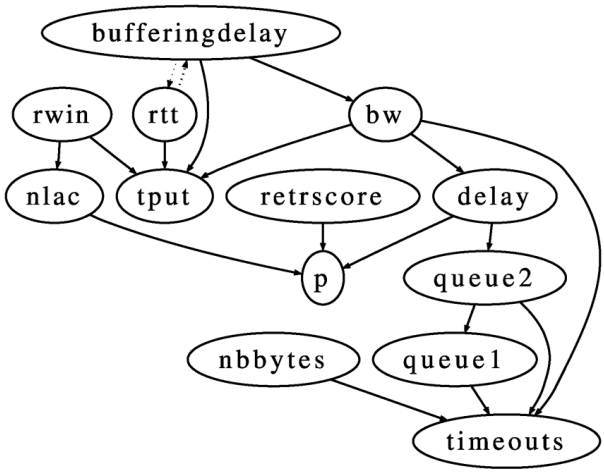


Fig. 16. Causal model inferred by the PC algorithm, with Z-Fisher criterion, for the emulated network traffic.

D.1.3. Result Log Linear. Figure 17 presents the causal model obtained with the PC algorithm and Z-Fisher criterion when applying a log-linear transform to the dataset presented in Table I. Based on Equation (1), applying a log transformation to the data could have could have brought the dependences closer to linearity, but the graph we obtain tends to disagree with this hypothesis. It can be observed that some independences seem to be better captured by this model, when compared to the independences implied by the original model (Figure 16). However, many of them are not in line with the mechanisms ruling the behavior of TCP. The throughput (*tput*) causing the receiver window (*rwin*) is one example, and the delay causing *p* is another. We can also notice the presence of a cycle ($tput \rightarrow rwin \rightarrow nlac \rightarrow tput$).

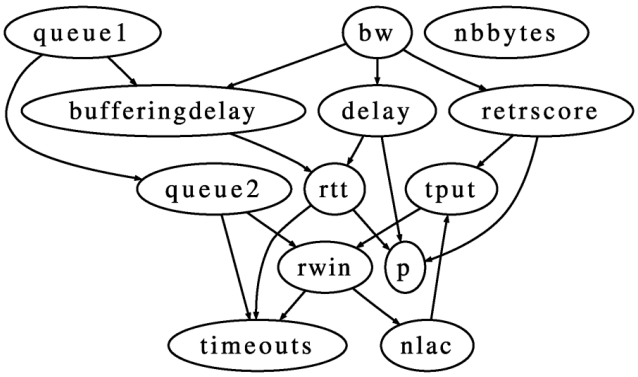


Fig. 17. Model for the network emulation data after applying log-linear transform, using Z-Fisher criterion.

D.1.4. Result IC Algorithm.* Figure 18 presents the model inferred using the IC* algorithm [Pearl 2009] on the emulated network dataset, summarized in Table I in Section 4.2. The graph output by the IC* algorithm may contain four types of edges:

(1) A marked arrow, signifying a directed path from X to Y, in the underlying model.
(2) An unmarked arrow, signifying either a directed path from X to Y or the presence of a latent common cause, L, of X and Y, in the underlying model.

(3) A undirected edge signifying the presence of a latent common cause of X and Y, in the underlying model.
(4) A bi-directed edge signifying any of the previously mentioned possibilities plus the possibility of a directed path from Y to X, in the underlying model.

The model we obtain shows a bi-directed edge between the *NLAC* and the *bufferingdelay* that gives no information on this dependence. However, it can be noticed that orienting this edge in one direction or another does not change the Markov equivalence class of the model and may be the reason why it cannot be oriented. Another property of this model is that all the edges going into *tput* are marked edges, showing that the parameters we observe are all direct causes of the *throughput* (*tput*), which supports our choice of intervention in Section 4.4.

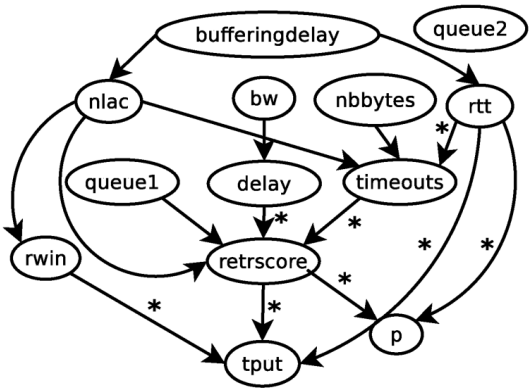


Fig. 18. Model for the network emulation data using the IC* algorithm.

D.1.5. Result FCI Algorithm. The FCI algorithm [Spirtes et al. 2001], by opposition to the PC algorithm, does not return a DAG (or more precisely an equivalence class of DAGs represented by a Partial Ancestral Graph (PAG)) but a Maximum Ancestral Graph (or more precisely an equivalence class of MAGs that can be represented by a PAG) with the following edges $\circ\!\!-\!\!\circ$, $\circ\!\!-$, $\circ\!\!\rightarrow$, $\longrightarrow$, $\longleftrightarrow$, $-$, with the following interpretations:

—There is an edge between X and Y if the corresponding parameters are conditionally dependent for any set of variables containing all selection variables[12] and a subset of observable variables.
—The presence of a tail means that the tail is present in every MAG of the equivalence class.
—The presence of an arrowhead means the arrow head is present in every MAG of the equivalence class.
—A $\circ$ edgemark means that there is at least one MAG where this egdemark is an arrowhead and one MAG where this edgemark is a tail.

Roughly, the presence of a undirected edge indicates the presence of a possible selection variable and a bi-directed edge the presence of latent variable.

Figure 19 presents the model inferred using the FCI algorithm with the Z-Fisher criterion.[13] As already observed, Z-Fisher does not seem to be able to correctly capture the dependences between the different parameters. If observing the subgraph $nlac \circ\!\!\rightarrow p \leftarrow\!\!\circ retrscore$, the presence of $\circ$ edgemarks in V-structure with a collider, p,

[12]Selection variables are unobserved parameters that values condition the observation of the sample (e.g., the connection was not aborted).
[13]In our work, we used the implementation from the R software Pcalg [Kalisch et al. 2012], which does not provide a solution for testing independences in the presence of non-linearity and non-normality.

is misleading but comes from the use of a criterion called possible-d-separation which extends the d-separation criterion presented in Section 3.2.1. As this criterion tends to consider a bigger set of possibilities, by having an inaccurate independence criterion, it leads to a less informative model.

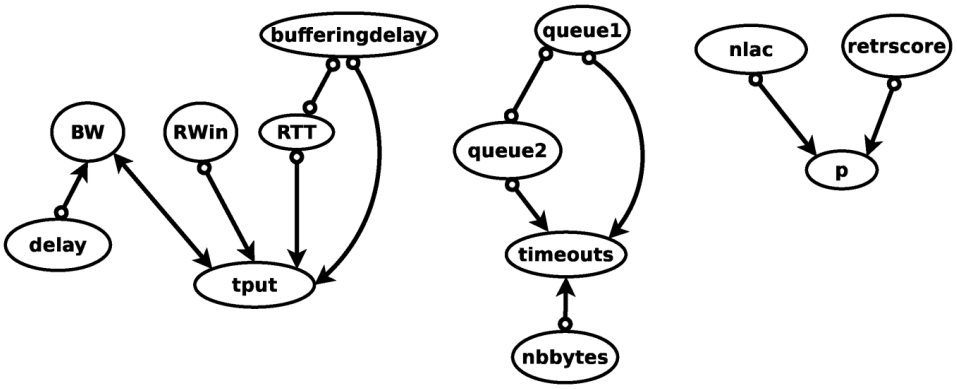


Fig. 19. Causal model inferred by the FCI algorithm and Z-Fisher criterion for the network emulation data.

D.1.6. Result kPC Algorithm. Figure 20 presents the model inferred by the kPC algorithm when applied to the emulated network scenario, which dataset is summarized in Table I. We can see two important negative points in this model. The first one being that, globally, this model gives very little information. Most of the parameters known to be impacting the throughput are independent one from another and the edges connecting them to the throughput are left unoriented. This last observation brings us to the second point, which comes from kPC algorithm orientation phase. The kPC algorithm highly relies on the non-linear and non-parametric regression. To orient an edge between two parameters found to be dependent, it tests the independence between the regressor and the residual for the two possible orientations. If one orientation leads to the independence of the residual on its regressor and the other orientation does not, then the edge is oriented accordingly. Otherwise, the edge is left unoriented. This model shows the difficulty that exists to model the non-linear dependence between two parameters.[14]

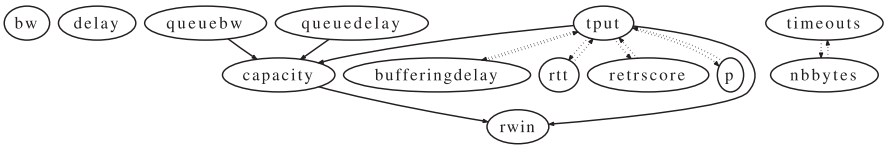


Fig. 20. Causal model inferred by the kPC algorithm for the emulated network scenario.

E. ADDITIONAL RESULTS FOR REAL FTP DATA

E.1. Causal Models

E.1.1. Result Using Z-Fisher Test. To illustrate how important it is to use the test best adapted to the nature of our data, which is non-linear and not normally distributed, we present in Figure 21 the model inferred by the PC algorithm using the Z-Fisher criterion instead of the KCI test. Most of the dependencies and orientations present in this model are incorrect, given our domain knowledge, the TCP mechanism, and the

[14]The kPC algorithm implementation from Tillman et al. [2009] originally uses support vector regression, but due to error in its implementation from the Spider toolbox, multi-ridge regression is used instead.

literature. The graph in Figure 21 shows the receiver window (*rwin*), the time of the day (*tod*) and the number of hops (*nbhops*) as parents of distance (*dist*) which is an exogenous variable. Similarly, the RTT (*rtt*) is an empirical cause of TCP throughput, while this model arrives at the opposite result. It seems that the Z-Fisher criterion fails to capture the dependencies between the TCP performance parameters.

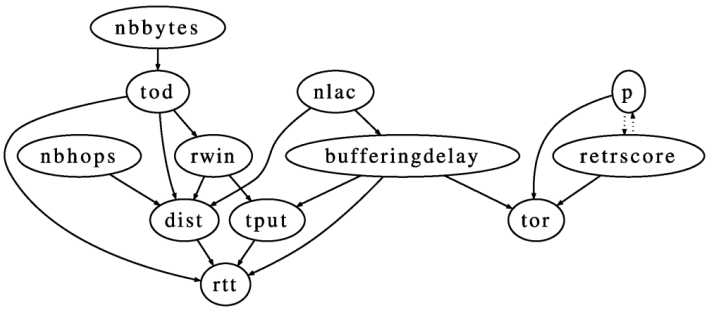


Fig. 21. Causal model inferred by the PC algorithm, with Z-Fisher criterion, for the real FTP traffic.

E.1.2. Result Log Linear. Figure 22 presents the causal model obtained with the PC algorithm and Z-Fisher criterion when applying a log-linear transformation. Based on Equation (1), this transformation of the data tries to obtain approximately linear dependencies.

Here, again, some of the causal dependences support the possible inaccuracy, even after log-linear transform, of the Z-Fisher criterion to capture the dependences between the parameters. Some of the direct parents of the throughput (*rtt*, *rwin*, *retrscore*, *p*) are the ones we expect to find even if some of the dependences are somehow counter intuitive (e.g., *rwin* $\rightarrow$ *nbhops*). These parameters are the ones used in PFTK model, showing its validity in terms of parameters directly influencing the throughput, and its limitation in modeling their dependences. The model also presents a causal dependence between the time of the day (*tod*) and the distance (*dist*), and the narrow link available capacity (*nlac*) and the distance (*dist*) that are oriented in a counter intuitive way. Note that the edges are part of a V structure entering a collider and cannot, then, be re-oriented without changing the Markov equivalence class to which the model belongs.

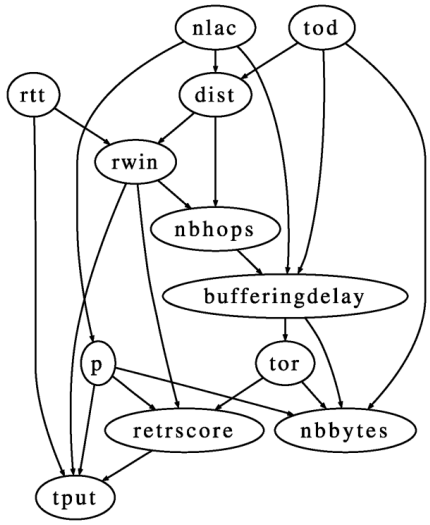


Fig. 22. Causal model inferred with PC and Z-Fisher when applying log-linear transform to the real FTP scenario.

E.1.3. Result IC Algorithm.* Figure 23 presents the causal model inferred by the IC* algorithm when applied to the real FTP dataset, summarized in Table II in Section 5.1. We can observe that all the edges are directed, apart from the one between *dist* and *nbhops*, and all the parents of the *throughput* show marked edges going into *tput*, excluding the possible presence of latent variables, apart from the one from *p* to *tput*. The presence of an unmarked edge means that the algorithm could not find whether there is a latent variable between the two parameters. In this case, it is our understanding of the TCP mechanisms that helps us discarding the possibility of a latent variable between the loss event probability (*p*) and the throughput (*tput*). The bi-directed edge between *dist* and *nbhops* cannot be oriented by the orienting method we use in the PC algorithm. In this case, using the weakly additive noise (WAN) model approach could have helped the orientation of this edge. Taking into account that *nbhops* is discrete, with few different values, and that the distance is a continuous variable, it would require adopting an accurate functional model for the non-linear regression of *nbhops* on *dist* and of *dist* on *nbhops*.

A last remark about the absence of latent variables in the model presented in Figure 23 concerns the time of the day parameter (*tod*). When we first started this study, we did not include the time of the day in our model. The model inferred by the PC algorithm, with the KCI+bootstrap method, presented unexpected dependencies and orientations, suggesting the presence of a latent variable, which was also indicated by the IC* algorithm. It is our domain knowledge that allowed us to reason about our model and add the time of the day (*tod*) variable in our system and obtain a consistent model.

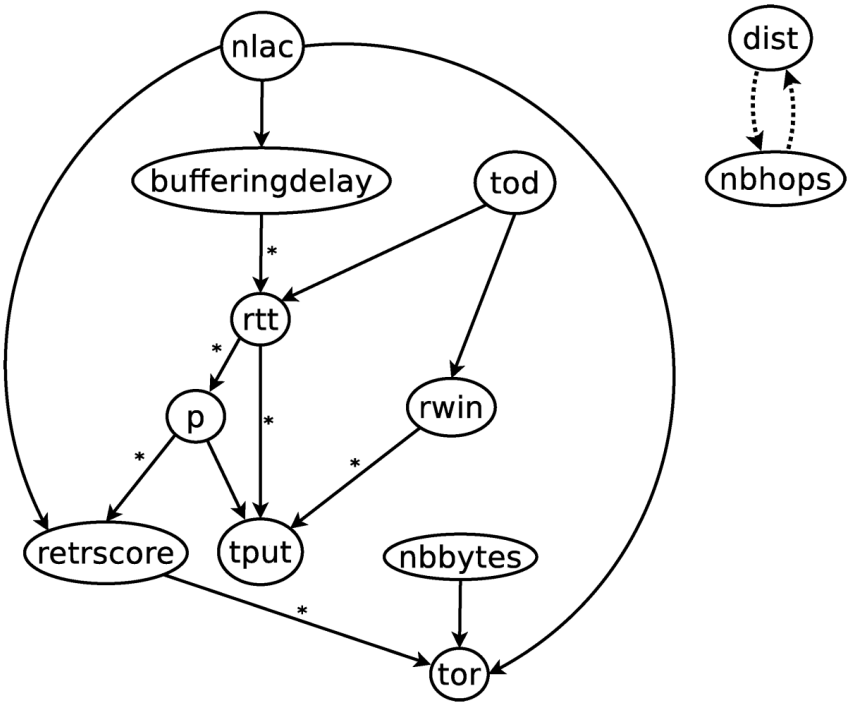


Fig. 23. Model for the real FTP data using the IC* algorithm.

E.1.4. Result FCI Algorithm. The results obtained when using the FCI with the Z-Fisher criterion are presented Figure 24. Here, the conclusions are similar to the ones concerning the model inferred by the FCI algorithm for the emulated network dataset,

as the Z-Fisher criterion is again used to test the independences between the different parameters.

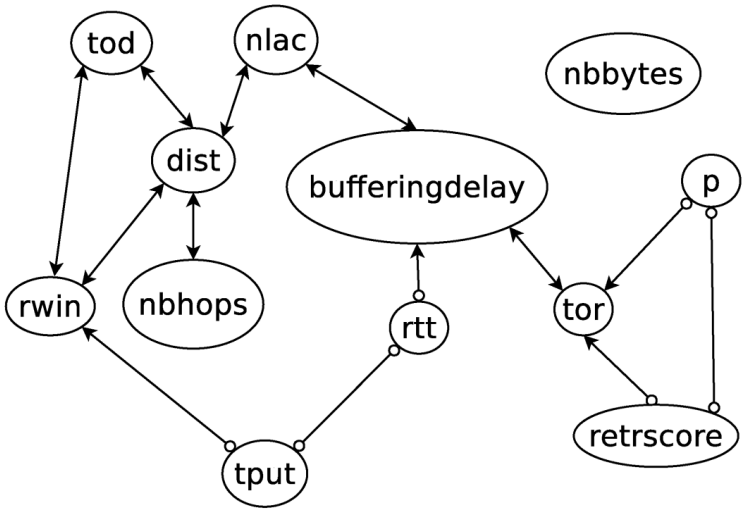


Fig. 24. Model obtained when using FCI algorithm with Z-Fisher criterion in the real FTP scenario.

E.1.5. Result kPC Algorithm. Figure 25 presents the model inferred by the kPC algorithm when trying to build a causal model of the real FTP scenario, whose dataset is summarized in Table II. Here, also, the algorithm gives a very uninformative model. Some of the parameters we know as impacting the throughput are found to be its direct parents, but the graph mainly shows them as being all independent. The interest in causal study lies in its ability to detect associations between the parameters that would be naïvely used as explanatory variable for the study of the target variable (the throughput in our case). In this case, the algorithm does not rely on Meek rules [Pearl 2009] to orient the edges of the skeleton. The use of non-linear regression under the assumptions of WAN models, allows the kPC to infer a DAG and not a PAG. Given these considerations, we cannot re orient the edge between *nlac* and *tod* while this would have given a more intuitive model. This example shows the difficulty, when working with real data, to capture non-linear dependences and its impact on methods that rely on their characterization.

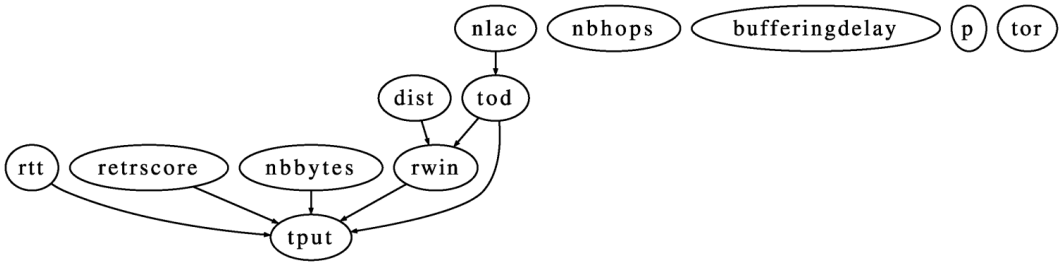


Fig. 25. Graphical causal model inferred by the kPC algorithm for the real FTP scenario.